HELIX PRODUCER USER'S GUIDE

Helix™ Producer 9.0 from RealNetworks®

Revision Date: 19 July 2002

CONTENTS

RealNetworks Producer for Helix User's Guide

INTRODUCTION

Welcome to the Helix™ Producer 9.0 from RealNetworks® User's Guide from RealNetworks. This manual shows you everything you need to know about Helix Producer—a program that allows you to convert audio and video into streaming media.

Using Helix Producer, anyone can easily create streaming media from a variety of sources. Convert from audio or video files, record directly from audio/video (A/V) devices, or use Helix Producer together with Helix Universal Server to broadcast and stream live content.

What is Helix?

Helix™ from RealNetworks is a universal digital media delivery platform. With industry-leading performance, integrated content distribution, advertising, user authentication, Web services support, and native delivery of RealMedia, Windows Media, QuickTime, and MPEG-4, Helix from RealNetworks is a robust digital media foundation that meets the needs of enterprises and networking service providers.

New Features for This Version

Helix Producer is completely new and redesigned for version 9. The following is a summary of some of the key new features:

- **New Graphic Interface**—gives you a more usable workspace
- **RealVideo 9 Encoding**—allows you to create the best quality video for any bit rate, especially at high-bandwidth rates
- **Encoding Jobs**—allow you to save and share your encoding settings, such as sources used, destinations encoded to, and other settings.

- **Multiple Destinations**—send the encoded output to more than one Helix Universal Server, or to a file and a server simultaneously.

- **Audience Templates**—allow you to save and share information about your target audience, such as codec used, target bit rate, and target frame rate.

- **Server Definition Templates**—allow you to save and share information about setting up a Helix Universal Server for live broadcasting.

- **Enhanced Live Broadcasting**—new broadcasting models, such as Push and Pull that allow you to better customize your Helix Universal Server connection.

- **Logging**—allows you to track important events that occur during encoding.

Conventions Used in this Guide

The following table explains the typographical conventions used in this guide.

Notational Conventions

Convention	Meaning
emphasis	Bold text is used for in-line headings, user-interface elements, URLs, and e-mail addresses.
terminology	Italic text is used for technical terms being introduced, and to lend emphasis to generic English words or phrases.
syntax	This font is used for fragments or complete lines of programming syntax (markup).
syntax emphasis	Bold syntax character formatting is used for program names, and to emphasize specific syntax elements.
variables	Italic syntax character formatting denotes variables within fragments or complete lines of syntax.
[options]	Square brackets indicate values that you may or may not need to use. As a rule, when you use these optional values, you do not include the brackets themselves.
choice 1|choice 2	Vertical lines, or "pipes," separate values that you can choose between.
...	Ellipses indicate nonessential information omitted from examples.

Additional Documentation Resources

In addition to this introductory guide, you may need the following resources, which are available for download at **http://service.real.com/help/library/encoders.html**:

- *Introduction to Streaming Media*

  Start with this guide if you are new to streaming media or RealNetworks products. Written for the beginning user, it explains how to put together a basic presentation using different production techniques.

- *RealNetworks Production Guide*

  This guide is the main reference manual for streaming media production. It expands on most of the topics presented in this introductory guide. Refer to it for instructions and tips on media production, as well as for complete information about using SMIL.

- *RealOne Player Scripting Guide*

  If you are a Web programmer, refer to this guide for instructions about using Javascript or VBScript with RealOne Player. Using these scripting languages, you can customize RealOne Player to turn it into your own Internet jukebox, for example.

- *Helix Universal Server Administration Guide*

  The basic reference for the Helix Universal Server administrator, this guide explains how to set up, configure, and run Helix Universal Server to stream multimedia. You need this guide only if you are running Helix Universal Server yourself. It is available at **http://service.real.com/help/library/servers.html**.

Technical Support

To reach RealNetworks' Technical Support, you can contact their Web site by choosing **Help>Technical Support** from the Helix Producer main menu.

The information you provide in this form will help Technical Support personnel respond promptly.

For general information about RealNetworks' Technical Support, visit this Web page:

- **http://service.real.com**

SYSTEM REQUIREMENTS AND INSTALLATION

This chapter shows you the hardware and software requirements for Helix Producer and how to install the product. Instructions are given for the following operating systems: Windows and Linux.

Windows Requirements and Installation

This section outlines the basic hardware and software requirements needed to run Helix Producer in a Windows environment. This section also gives you step-by-step instructions for installing the product onto a Windows machine.

Windows System Requirements

The following table lists the minimum and the recommended requirements for using Helix Producer on Windows.

Windows Requirements

Requirement	Minimum	Recommended
CPU	400 MHz	800+ MHz
RAM	32 MB (file to file encoding) 96 MB (live broadcasting)	256 MB
Operating System	Windows NT 4, SP 6 Windows 2000 Windows ME Windows 98 SE/XP	Windows NT 4, SP 6 Windows 2000
Hard Disk space (software)	20 MB	
Hard Disk space (data)	500 MB	1 GB
Color Display	16-bit	24-bit (TrueColor)
Sound Card	16-bit sound card or better	

Installing the Windows Version

Follow this procedure to install the product onto a computer running Windows.

➤ To install Helix Producer onto Windows:

1. Download the installation program.

2. Close any other applications that may be open and double-click on the installation program icon.

   The installer begins by unpacking the required files for Helix Producer installation. When complete, the installer opens to the License Agreement page.

3. Read the terms and conditions carefully, and select **I accept the terms in the licence agreement** if you agree. You can click **Cancel** at any time to abort the installation.

4. Click **Next** to continue.

   The Serialization page opens.

5. Enter your name, company name, and serial number you received when you purchased Helix Producer.

6. Click **Next** to continue.

   The Install Options page opens.

7. The default install directory that Helix Producer will install to your computer is listed. You can change this default install directory by clicking the **Browse** button and selecting the new path.

8. You also have the options to select whether a shortcut to Helix Producer will be created on the desktop and/or on the Quick Launch toolbar, and whether command line tools will be added to your PATH.

   > **Warning!** You must have enough disk space and write access to the location where you intend to install Helix Producer. If not, you will not be able to continue the installation.

9. Click **Next** to continue.

   The install program installs all necessary files and opens the Finishing Up page.

10. Choose whether you want to launch Helix Producer and/or the Readme file when you exit the install program.

11. Click **Finish** to exit.

Linux Requirements and Installation

This section outlines the basic hardware and software requirements needed to run Helix Producer in a Linux environment. This section also gives you step-by-step instructions for installing the product onto a Linux machine.

Linux System Requirements

The following table lists the minimum and the recommended requirements for using Helix Producer on Linux.

Linux Requirements

Requirement	Minimum	Recommended
Version	Linux 2.2 and 2.4 with glibc 2.1 or greater	
CPU	400 MHz	800+ MHz
RAM	32 MB (file to file encoding) 96 MB (live broadcasting)	256 MB
Hard Disk space (software)	20 MB	
Hard Disk space (data)	500 MB	1 GB

Installing the Linux Version

Follow this procedure to install the product onto a computer running Linux.

➤ To install Helix Producer onto Linux:

1. Download the installation program.

2. Make sure your CD drive is mounted properly, if installing from a CD.

3. Copy the installer to your hard drive if you are getting it from a CD.

4. Start the Helix Producer installer by entering:
   ```
   sh helix_producer_plus_9_linux.bin
   ```
   The installer extracts the necessary files to the directory where the installer is located.

The install program installs all necessary programs and files.

When complete, the installer shows the Software License agreement.

Other Basic Requirements

In addition to normal hardware and software requirements, it is recommended that you have the following products:

- the latest version of the RealOne Player
- speakers or headphones connected to your sound card

For recording from a media device, any of the following connected to your computer:

- video capture card, if encoding live video
- S-VHS, Digi-Beta, or Beta-SP video player
- video camera
- microphone

STREAMING MEDIA BASICS

This chapter introduces you to streaming media and how Helix Producer creates streaming media. It gives you a brief look at how streaming works, the different types of streaming media that you can create with Helix Producer, and the different RealNetworks products that you use when streaming.

What is Streaming Media?

Streaming media is also known as RealMedia, or separately as RealAudio® or RealVideo®. Before the advent of streaming media, your audience had to wait for your media file to download from the Internet or a network server in order to experience it. With streaming media, your audience can see and hear your media almost instantly.

A streaming clip consists of small packets of information that are sent over a network connection. The user receives these information packets in a "stream" and, using a player, experiences your media piece by piece. Thus, the streaming is almost invisible to the user. The process is similar to viewing a film; each frame of a filmstrip is like each data packet. When they are pieced together by a projector, you see the film as one continuous piece.

How is Streaming Media Created?

Helix Producer creates streaming media data packets by a process called "encoding." During encoding, the source media is transformed into streaming media using "codecs" (compression/decompression algorithms). The entire process is summed up in the following steps:

1. Helix Producer receives the source media as a file or live audio/video.

2. Helix Producer uses a codec to compress the media source's data into packets.

3. The data packets are streamed via the Internet or network to the user.

4. At the user's end, the same codecs are used to piece back the media so that he or she can play it.

While the steps involved in encoding streaming audio and video are similar for you, for Helix Producer there are some basic differences between the media types.

Encoding audio is much simpler than encoding video. A basic, streaming audio clip is created by using a audio file or live audio source as the input. Helix Producer uses various audio codecs to convert your standard audio into a format that can be streamed.

A more complex task for Helix Producer is converting standard video into streaming media. A RealVideo clip is created by converting a video file or by capturing from a video source, such as a video camera or video cassette player, to your computer via a video capture card. Helix Producer converts different attributes of the video—such as frame rate, type of motion, and size of the image—into a RealVideo clip using a video codec. Plus, if the video includes audio data, that must also be converted using the audio codecs.

Targeting Audiences

Before Helix Producer can compress the input media data, it needs to know something about the audience that you will be targeting. An audience is defined by the bit rate at which they can connect. For example, a person using a 56 kbps dial-up modem to connect to your stream is a member of the 56K Modem audience. If you target only one audience, you are targeting a single bit rate.

Single bit rate streaming

Real Media   Single Bit Rate   Player
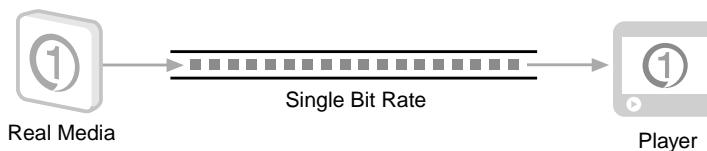
Since compressing data loses some information, picking the correct audience is key to deciding how much of your source's data you keep. With Helix Producer's SureStream technology you can reach the widest possible audience, and provide all users with the best listening and viewing experience optimized for their bandwidth.

SureStream streaming
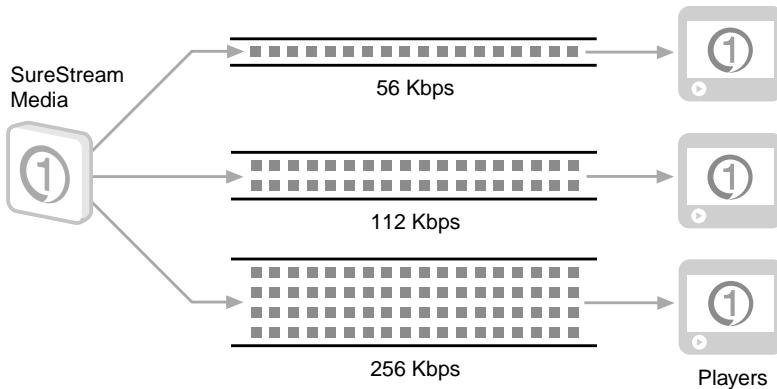
There are advantages to using SureStream. You can create a single RealMedia clip recorded for multiple target audiences, or you can create a clip that will automatically switch to a lower bandwidth during poor network conditions. SureStream RealMedia files can combine several different streams that take advantage of these features.

For example, you can record a video clip for 56 kbps, 112 kbps, and 256 kbps audiences, and RealOne Player will automatically use the correct stream based on the user's connection speed. All streams are contained within a single RealMedia file.

About Other RealNetworks Products

Helix Producer is a member of the RealNetworks family of software products. Three components works together to stream your media:

- Helix Producer—to create streaming RealMedia clips (such as RealAudio or RealVideo). This process is also called *encoding*.

- Helix Universal Server—to stream the RealMedia clips.

- RealOne Player—for the audience to use to play the streamed RealMedia clips.

The following diagram illustrates how these RealNetworks products work together.

RealNetworks software products

Server

Helix Producer

RealOne Player
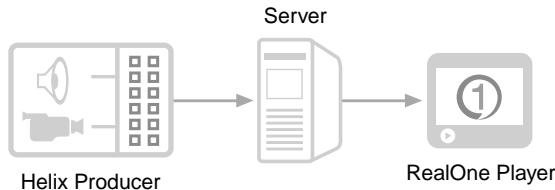
Encoding and SMIL

You are the person who creates the streaming media to be used with RealNetworks products. As the "content creator," you will use Helix Producer. This product is used to encode audio and video into RealMedia that Helix Universal Server can then stream.

You can also create a Synchronized Multimedia Integration Language (SMIL) file to synchronize several clips within a presentation. SMIL is an extensible mark-up language, designed to be the standard markup language for timing and controlling streaming media clips. A SMIL file coordinates the layout and playing of two or more media clips in parallel (simultaneously) or in sequence. For an in-depth look at SMIL and how to use it to create RealMedia productions, refer to the *RealNetworks Production Guide*.

Helix Universal Server

Just as a Web server delivers pages to Web browsers over the Internet, Helix Universal Server serves streaming media clips to your audience. It enables users to stream the media clips rather than download them. By streaming the content, the user can begin to watch the clip almost immediately and doesn't have to wait for the entire file to be downloaded.

There are two different ways to deliver your content. You can deliver it on-demand or broadcast it live. On-demand means that you encode your RealMedia content before your audience needs it. Live broadcasting is when you are encoding the RealMedia at the same time your audience is playing it. For both types of delivery, you can use Helix Universal Server in conjunction with Helix Producer.

Client Software

A client such as RealOne Player plays the streamed media. Helix Universal Server also supports streaming content to QuickTime and Windows Media Players.

GETTING STARTED WITH HELIX PRODUCER

This chapter gives you a quick look at how to use Helix Producer to encode streaming media. It will introduce you to the Helix Producer interface, and then it will guide you step-by-step through some basic encoding tasks. You will learn the basics on encoding a RealMedia file from a source file, and you will learn the basics of capturing and encoding live RealMedia.

> **Note:** The information in this chapter is intended for the beginning user of Helix Producer. It is not meant as a substantive guide to using the product, only a starting point that introduces you to the product and shows you the general steps involved in encoding streaming. The remainder of this manual will give you more details about the steps involved.

Before You Begin: Learning About Helix Producer

The first step in learning about a program is to start it and familiarize yourself with the interface. This section introduces you to the Helix Producer main window, describing each section. And it shows you how to open the various palettes that you will use when preparing to encode streaming media.

Starting Helix Producer

The first step is to open the product and familiarize yourself with the Helix Producer main window.

➤ To open Helix Producer:

• From the Windows Start Menu, select **Programs>**Helix Producer.

The Helix Producer main window opens.

The Helix Producer Main Window

The main window is your "workspace." Here you will prepare to encode by adding inputs, outputs, and other settings, and you will also view the actual encoding of the RealMedia.

The following is a list of some of the important sections within the main window:

Input Audio/Video

This section of the main window includes an audio meter that allows you to view the audio gain (volume) from the input audio. Also, the Input video screen allows you to view the source video.

Output Audio/Video

This section of the main window includes an audio meter that shows the output audio gain. Also, the Output video screen allows you to see how the encoded video will look to your audience.

Input

This section of the main window is where you decide what the source for your input will be. You can select either an input file or an audio/video device.

For more information on setting up the input, see Chapter 5: Defining the Input for an Encoding Job on page 27.

Output

This section of the main window is where you decide where the encoded content will be sent. You can choose to create a RealMedia file and/or you can choose to send the content directly to a Helix Universal Server for a live broadcast. Each of the destinations that you set up will be listed in the Output Manager.

For more information on setting up the output, see Chapter 6: Defining the Output For an Encoding Job on page 35.

Job Manager

This section of the main window is where you can control the different encoding jobs that you have saved on your computer.

An encoding job is the container for all the settings you have selected for the encoding process: the source used for the input, the destinations used for the

output, and other settings that determine how the RealMedia content is created.

You'll learn more about setting up jobs and the Job Manager in Chapter 4: Setting Up an Encoding Job on page 21.

Step by Step: Creating a RealMedia File

The following are the basic steps involved in creating a streaming RealMedia file that your audience can view on-demand:

Step 1: Start with a Media Source—File or Live Input

Step 2: Select a File Destination for the Output

Step 3: Encode Your Source into RealMedia

Step 4: Move Your Clip to a Server

Step 5: Play the Clip

Step 1: Start with a Media Source—File or Live Input

Helix Producer creates streaming media from two kinds of sources: files or devices. Either of these sources can contain audio-only content or video and audio content together. You use the Input section of the main window to choose the source for the input.

➤ To select a source:

1. In the Input section of the main window, select either:

   • **Input File**
   • **Devices**

2. If you select a file for the input:

   a. Enter the media file's full path and file name.
   b. Or click **Browse** and find the media file.

3. If you select a device for the input:

   a. Select an audio device for encoding the audio portion of your input.
   b. Select a video device for encoding the video portion of your input, if you are encoding video.

You'll learn more about setting up a source in Chapter 5: Defining the Input for an Encoding Job on page 27.

Step 2: Select a File Destination for the Output

In this step you designate what the output RealMedia file will be named and where it will be located when the encoding is complete.

➤ To select a File Destination:

1. In the Output section of the main window, click the **File Destination** button.

2. Change the directory to the location where you want the file to be saved.

3. Enter a name for the RealMedia file that will be created.

    > **Note:** If you selected a media file as your source, a default name will be created that you can keep or change.

4. Click **Save**.

You'll learn more about setting up a file as a destination in "Creating a RealMedia File" on page 36.

Step 3: Encode Your Source into RealMedia

Now that the input and the output are set, you can encode the RealMedia file. The encoded file will be created using default settings.

➤ To encode a RealMedia file:

1. Click the **Encode** button.

    The encoded output video will appear in the Output video screen.

2. If you are encoding a file, you can wait for the file to finish encoding, or you can stop early by clicking the **Stop** button.

3. If you are encoding from a device, click the **Stop** button.

For more information on encoding, see "Starting an Encoding Job" on page 73.

Step 4: Move Your Clip to a Server

You can now use a server to deliver your RealMedia file to your audience. You use a Helix Universal Server to stream your clip over the Internet (or to an Intranet). Helix Universal Server is a separate software product, created by RealNetworks, that allows you to stream live or recorded RealAudio and RealVideo files to anyone using a RealOne Player. The server can work with

Web servers to stream media over the Internet or to users on a corporate Intranet.

➤ **To move your file to a server:**

- Copy the `ondemand.rm` clip you created in the preceding step, from its current location to the Helix Universal Server `Content` directory.

   On Windows NT, the path is `C:\Program Files\Helix Server\Content`.

   On UNIX, the path is `/usr/local/Helix Server/Content`.

For more information on how you can obtain and use a Helix Universal Server, go to the RealNetworks Web site at **http://www.realnetworks.com** and look for the links to this product.

Step 5: Play the Clip

Once your audience has connected successfully to your server to view the streaming media content, they will use a player to view it. A RealOne Player is all they need. The RealNetworks Web site (**http://www.realnetworks.com**) has more information about each version and how to install and use them.

Step by Step: Creating a Live Broadcast

The following are the basic steps involved in creating and streaming a live RealMedia broadcast that your audience can view as it is being encoded:

Step 1: Start with a Media Source—File or Live Input

Step 2: Select a Server Destination for the Output

Step 3: Encode Your Source to a RealMedia Broadcast Stream

Step 4: Play the Clip

Step 1: Start with a Media Source—File or Live Input

Helix Producer creates streaming media from two kinds of sources: files or devices. Either of these sources can contain audio-only content or video and audio content together. And either source can be used for a live broadcast, even though usually live audio and video is captured for a live broadcast.

You use the Input section of the main window to choose the source for the input.

➤ To select a source:

1. In the Input section of the main window, select either:
   - **Input File**
   - **Devices**

2. If you select a file for the input:
   a. Enter the media file's full path and file name.
   b. Or click Browse and find the media file.

3. If you select a device for the input:
   a. Select an audio device for encoding the audio portion of your input.
   b. Select a video device for encoding the video portion of your input, if you are encoding video.

You'll learn more about setting up a source in Chapter 5: Defining the Input for an Encoding Job on page 27.

Step 2: Select a Server Destination for the Output

You can only use a Helix Universal Server to stream a live broadcast with Helix Producer. In order to do this, you must create a Server Destination. Your system administrator will have all the information you need to create the Server Destination.

➤ To create a Server Destination:

1. In the Output section, click the **Server Destination** button.

   The Server Destination dialog opens. As mentioned above, these settings will have to be filled out with help from your system administrator.

2. Enter a name for the RealMedia file that will be used by the server.

3. Enter the name for the Server.

4. Enter settings that allow Helix Producer to connect to Helix Universal Server.

5. Click **OK** to save the destination.

You'll learn more about setting up a server as a destination in "Broadcasting Live to a Server" on page 37.

Step 3: Encode Your Source to a RealMedia Broadcast Stream

Now that the input and the output are set, you can encode the RealMedia live broadcast. The encoded RealMedia content will be created using default settings.

➤ To encode a RealMedia file:

1. Click the **Encode** button.

   The encoded output video will appear in the Output Video Viewer.

2. If you are encoding a file, you can wait for the file to finish encoding, or you can stop early by clicking the **Stop** button.

3. If you are encoding from a device, click the **Stop** button.

For more information on encoding, see "Starting an Encoding Job" on page 73.

Step 4: Play the Clip

Once your audience has connected successfully to your server to view the streaming media content, they will use a player to view it. A RealOne Player is all they need. The RealNetworks Web site (**http://www.realnetworks.com**) has more information about each version and how to install and use them.

SETTING UP AN ENCODING JOB

This chapter describes how to set up a job using the Helix Producer main window if you are using the Windows operating system. You will learn what an encoding job is, how to create and save jobs, and how to use multiple jobs at once.

About Encoding Jobs

The encoding job (or "job") is a file that you use to show Helix Producer how to encode RealMedia. A job contains all of the settings for the different aspects that make up your encoding session. On a basic level, when you set up a job, you are defining what the input is, how the input is encoded into streaming media, and where the encoded output will be sent. The following diagram illustrates this flow:

The basic structure of a job
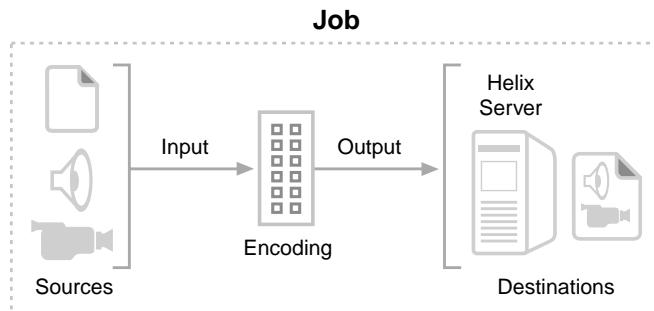
The following is a list of the different parts of a job.

- **Input**—the source (media file or live audio/video) that you will use to encode RealMedia.

- **Encoding Job Settings**—settings that show Helix Producer how to interpret the input and then encode into streaming media. Some

examples of job settings are: whether or not to analyze data before encoding, which filters to use to process video data, what specific audience (or audiences) will be targets, and what information about the encoded clip will be shown to your audience.

- **Output**—the destinations (RealMedia file and/or live broadcasts to servers) for the encoded RealMedia.

All of these parts that make up a job are described in greater detail throughout the remaining chapters. But first, this chapter shows you how to create and save a job for later use, and it describes the benefits of using multiple jobs.

Creating a New Job

The first step in encoding with Helix Producer is to create a new job.

You can let Helix Producer create it automatically by setting the input or an output destination. Or you can manually create a new job to work with.

When you manually create a new job, an untitled job appears in the Job Manager. This untitled job does not have an input or output defined and it uses default settings for encoding.

➤ To create a new, untitled job:

1. From the Helix Producer main window, select the **File** menu.

2. From the **File** menu, select **New Job**.

   A new job appears in the list of jobs at the bottom of the window. This list is called the Job Manager.

About the Job Manager

The Job Manager keeps track of the encoding jobs that you are using during the current session. It also allows you to quickly control more than one job at a time. See "Using Multiple Jobs" on page 24 for more information.

Below is a description of the different columns in the Job Manager:

Job—the name of the job.

Status—gives you a description of the status of the job, such as Ready, Not Ready, Analyzing, Encoding, and Done.

Start—the time that the job began encoding.

End—the time the job stopped encoding.

Saving and Sharing a Job

Once your job is created, you can define the different components that make up a job. After that, you will want to save your job so that you can use it at a later time or share it.

➤ To save a job to a file:

1. From the Helix Producer main window, open the **File** menu.

2. From the **File** menu, select **Save Job As**.

    The Save As dialog opens.

3. Browse to the directory where you want to save the job.

4. Type in the name for the job file and click **Save**.

    The file will be saved with a .rpjf extension.

Saving a job creates a job file with a .rpjf extension that is stored on your computer. This job file can be modified in two ways. You can open the job (see "Opening Existing Jobs" below) and change its settings using the Helix Producer interface. Or you can modify the job file using an XML editor or text editor once you are more familiar with the other components and settings that make up a job. See Appendix B: "Job File Reference" for more information on editing the job file.

Using Jobs as Templates

You can create jobs, save them, and use them as templates. Create jobs for different encoding situations that frequently arise and then make incremental adjustments to the job.

For example, if you often encode a list of files into RealMedia for a modem audience, you can save a job with a dummy file input that you can modify later. Or if you need to broadcast live video over the Internet periodically but to different servers, you can create a job with all the other settings and just modify the destination. Thus, your broadcast is quickly up and running each time.

> **Tip:** If you want to save the current job's settings as the default settings, select **Settings>Default Settings>Save Current as Default** from the menu. You can always restore the original

default settings by selecting **Settings>Default Settings>Restore Original Default**

Sharing the Job File

Once you have saved your job, you can also share it. This means that you can give the job file (the .rpjf file) to another user. He or she can open it in his or her version of Helix Producer and use it to encode RealMedia. If you have set the input and output destinations, the new user needs access to the same sources and destination locations that you defined in the job.

Opening Existing Jobs

You don't have to create a new job every time you use Helix Producer. If a job has been previously created, or if you have been sent a job by a colleague, you can open it and add it to the Job Manager.

➤ To open an existing job:

1. From the Helix Producer main window, open the **File** menu.

2. Select **Open Job**.

   The Open Job File dialog opens.

3. Browse to the job you want to use and click **Open**.

   The job appears in the Job Manager and its settings appear in the main window.

Using Multiple Jobs

One of the more powerful features of Helix Producer is that you can create and control multiple jobs at the same time. The Job Manager is used to show the jobs that have been added and are waiting to be modified or encoded.

This section shows you some of the benefits of using multiple jobs and some tips on how to use them effectively.

Batch Encoding

Setting up more than one job to encode one after the other is called "batch" encoding. Batch encoding is most useful when you have a number of files that you need to encode using virtually the same settings.

➤ To encode a batch of jobs:

1. Create multiple jobs and add them to the Job Manager.

> **Tip:** To create a number of jobs with the exact same settings, select a job and drag-and-drop media files into Helix Producer. A new job is created for each file with the same settings as the selected job. You can also duplicate a job by right-clicking on a job and selecting **Duplicate** from the menu.

2. Select the batch of jobs that you want to encode.

   To do this, select each job by using the CTRL key while selecting, or you can select a group of jobs with the SHIFT key.

3. Encode the batch of jobs by clicking the **Encode** button.

   Each job that is in the batch will encode in series. This means that only one job will encode at a time.

For more instructions on encoding, see the chapter entitled "Starting and Monitoring an Encoding Job" on page 73.

Multi-Tasking Between Jobs

When you multi-task, you don't have to wait for a job to finish encoding before you can continue working with Helix Producer. While one job is encoding, you can add a job or create a new job and begin defining the settings for the different components of the job.

DEFINING THE INPUT FOR AN ENCODING JOB

This chapter describes how to define what the input will be for your encoding job. You will learn what you can use as the source for the input and how to adjust the input video.

Overview: Input and Sources

The input for a job is a media source that will be encoded into streaming media by Helix Producer. Whatever source you want to use will become the input for the job. The following diagram shows you three different jobs, each with a different type of source for the input.

Sources used as inputs
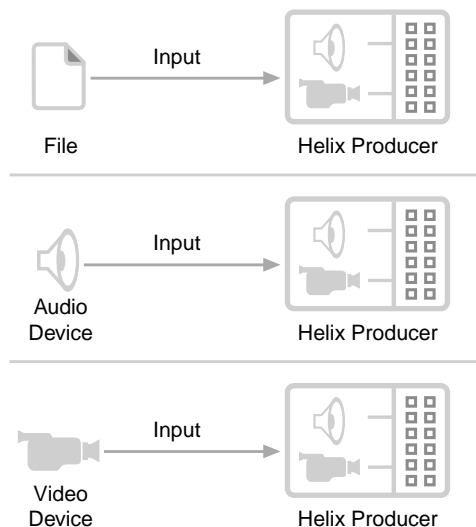
This section discusses the different types of sources you can use for an input, shows you how to add an input to a job, and shows you how to use video filters to adjust the input's video.

Setting the Source for the Input

There are two types of media sources that you can use to create streaming media with Helix Producer. You can use either files or an audio/video capture device.

This section is an introduction to these sources that you can use when you set up an encoding job with Helix Producer.

Using a File as the Input

Source files are pre-existing media files that are compatible with Helix Producer and can be encoded into RealMedia. These files are generally created with other media-generating tools.

➤ To add a file as the input:

1. On the Helix Producer main window, select **Input File** in the Input section.

2. Click the **Browse** button to locate the file you want to add as the input.

3. Change to the folder where the media file is located, select it, and click **Open**.

   The path and file name of the media file is listed in the Input File field.

   A default destination file appears in the Output section using the input file's name with a .rm extension.

You can use a variety of media file as sources for the input. The following lists are based on software you have installed on your computer.

Platform-Independent File Formats

The following file formats can be used as sources on any platform supported by Helix Producer. No other software needs to be installed to use these formats.

- Uncompressed AVI
- Uncompressed QuickTime 3, 4, and 5
- WAVE audio files

DirectShow Supported File Formats

The following file formats can be used as sources only on the Windows operating system. At a minimum, you must have DirectX 6 installed. Some

formats require that DirectX 8 is installed. For the latest version of DirectX, go to **www.microsoft.com** or use the link provided at **www.realnetworks.com**.

- Compressed AVI (.avi)
- AIFF (.aif, .aifc, .aiff)
- Moving Pictures Experts Group (.mpg, .mpeg, .m1v, .mp2, .mp3, .mpa, .mpe, .mpv2, .m3u)
- Audio for Windows (.wav)
- QuickTime 2 Content (.mov, .qt)

Using Live Audio/Video as the Input

Another source that you can use for the input is live audio and/or video. To send audio and video directly to your computer, you will need the following:

- a media device (such as a VCR, video camera, or microphone)
- for capturing live audio, a sound card
- for capturing live video, a video capture card, unless you can connect a video device directly to your computer via an IEEE -1394 (Firewire, iLink), or a USB port

➤ To add an Audio device as the input:

1. In the Helix Producer main window, select **Devices** in the Input section.

2. Select the audio device for your computer from the pull-down list (for example, **SoundBlaster**).

3. If you want to adjust the sound recording capabilities of your audio device, click the **Settings** button next to the listed audio device.

   A menu appears listing available settings dialogs for your audio device.

4. Select the **Recording Mixer** menu option.

   The Recording Control window opens. In this window you can specify how audio is mixed from different audio sources.

5. Select which recording inputs to use by choosing the **Select** box for each input.

6. Adjust the sound level by moving the sliders up or down.

   If the volume is too high, the recorded sound may be clipped and appear distorted. If the volume is too low, it will be difficult to hear. Use the

audio level meter in the Helix Producer window to monitor the level during encoding.

7. Close the Recording Control window and return to the Helix Producer.

➤ **To add a Video device as the input:**

1. On the Helix Producer main window, select **Devices** in the Input section.

2. Select the video device for your computer from the pull-down list.

3. If you want to adjust the video recording capabilities of your video capture device, click the **Settings** button next to the listed video capture device.

   A menu appears listing available settings dialogs for your video capture device.

   > **Note:** Every video capture device will have a unique way of modifying how the video is captured. The rest of this procedure will explain a few common settings. Consult your device's manual for more information.

4. Select the video dimensions, in pixels. This will determine how large your input video will be. Some common sizes are: 160 x 120, 320 x 240, and 640 x 480.

5. Select the video format. This is how the capture card converts the video into digital video. Some common formats are: RGB, YUY2, BTYUV, YUV9, and YUV12.

6. Select the video source. Some video capture devices allow you to plug in various sources, such as an S-video input from a video camera, a cable television cable, or a web cam.

Adjusting Input Video

Once an input is added to a job, Helix Producer gives you a variety of filters to improve the quality of video that comes from your source. You can crop the input video to a size more to your liking, and you can apply an array of filters to the input video. The rest of this section shows describes these methods.

Cropping the Input Video

Helix Producer's cropping feature allows you to record any portion of a video image that you wish. When you crop video, you reduce the amount of data that needs to be encoded, while also removing unwanted image area.

Your input must contain video to use cropping, and the smallest size that you can crop an image to is 32 x 32.

➤ To crop an input video image:

1. From the main window, click the **Video Filters** button.

   The Video Filters palette opens.

2. Select the checkbox labelled **Cropping**.

   Yellow lines appear on the preview section of the Video Filters dialog.

   > **Note:** The part of the video region that these yellow lines outline is the part of the video that will be encoded.

3. Using the mouse cursor, select and drag these yellow lines to crop the input video to the desired size.

4. You can also type in the **Left**, **Top**, **Width**, and **Height** dimensions of the desired video size in pixels.

Using the Black Level Correction Filter

The Black Level Correction filter is used to adjust the brightness of the input, effectively making black areas in the video appear "blacker." This filter is useful if the video appears washed out, without much saturation.

➤ To turn on the Black Level Correction filter:

1. From the main window, click the **Video Filters** button.

   The Video Filters palette opens.

2. Select the checkbox labelled **Black Level Correction**.

   The Black Level Correction filter is enabled.

Using the Deinterlace and Inverse Telecine Video Filters

The Deinterlace filter removes "artifacts" that can be present when encoding digitized NTSC or PAL formatted video. These artifacts are horizontal lines seen surrounding a moving object.

Most movies are filmed at 24 frames per second (fps). When a film is converted into NTSC video at 30 fps, fields are alternately duplicated and shared. The film-to-video conversion process is called telecining. The Inverse Telecine filter removes redundant fields added during a telecine process, improving the quality of the frames that are encoded.

The Inverse Telecine filter is only used when encoding input video with a frame rate of 30 fps.

➤ **To turn on the Deinterlace and Inverse Telecine filters:**

1. From the main window, click the **Video Filters** button.

   The Video Filters palette opens.

2. Select the checkbox labelled **Deinterlace/Inverse Telecine**.

3. You have two options:

   a. Select **Automatic**, and Helix Producer will use these filters if the input looks like it has artifacts or it looks like it was created with a telecine conversion.

   b. Select **Manual**, and then select either filter that you want to be applied.

   When selected, the filters are enabled.

Using the Video Noise Reduction Filter

Video noise appears as static in your input video. This noise can appear in video for a variety of reasons: poor capture cards, cameras, or storage. The noise filter removes this noise from the input video prior to encoding.

It is recommended that you do not use the video noise filter unless your video has some distortion or static. Using this filter (especially the High setting) will degrade the quality of undistorted video.

➤ **To adjust the Video Noise Reduction filter:**

1. From the main window, click the **Video Filters** button.

   The Video Filters palette opens.

2. Select the checkbox labelled **Video Noise Reduction**.

3. You have two options:

   a. Select **Low**, and Helix Producer will filter out some video noise.

b. Select **High**, and Helix Producer will filter out a significant amount of noise. As mentioned above, the High setting will degrade undistorted video.

DEFINING THE OUTPUT FOR AN ENCODING JOB

This chapter describes how to define the destinations for the output of your encoding job. You can send the encoded output to be saved as a RealMedia file for on-demand streaming, or to a Helix Universal Server for a live broadcast. You will learn in this chapter how to set up both types of destinations. And you will learn how to set information about the output clip that will be sent to your audience.

Overview: Output and Destinations

The output for a job is the encoded media created by the Helix Producer from the input media source. To define where the output will go once it is encoded, you specify the *destinations* for the encoded media. There are two types of destinations that you can define for an output: RealMedia files and Helix Universal Servers. A file is used when you are encoding for on-demand streaming. A server is used when you are encoding a live broadcast.

Destinations for a job's output
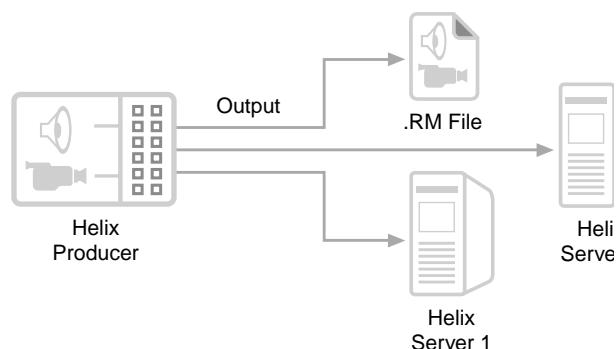
You can specify multiple destinations for the output, all of which will receive the encoded media. From the main window, you will see a list of destinations already defined for a job appearing in the output list.

Each destination you add to the output is independent. If there is a problem that stops encoded data from reaching one destination, the other destinations still receive the data. For example, you can add a server and a file destination. If the connection to the server is lost, Helix Producer continues to send the output to the file destination.

Creating a RealMedia File

You can use a RealMedia file as a destination, telling Helix Producer to take the encoded output and save it as a RealMedia file. A RealMedia file is a saved file that can be published later to a Helix Universal Server and then streamed to an audience. This method of encoding a file for later use is termed *on-demand* encoding. In other words, your audience can experience your content when they demand it. On-demand encoding is the primary use for using a RealMedia file as a destination.

A secondary use for using a RealMedia file as a destination is to archive a live broadcast. For this method, you set up both a Server Destination and RealMedia file as destinations. The exact encoded output that is sent to the server is also saved as a file in real time. Thus, your live broadcast is stored in a permanent form and can be streamed on-demand after the broadcast is complete.

Another benefit to encoding a RealMedia file is that you can view your content after it has been encoded. And if either the encoded audio or video does not meet your expectations, you can adjust encoding settings and re-encode.

➤ To define a RealMedia file as a destination:

1. From the main window, select **File>Add File Destination**.

   The Save As dialog opens.

2. Browse to the directory where you want the RealMedia file to be located.

3. Enter a name for the file. The extension is automatically provided by Helix Producer.

   > **Note:** Normal RealMedia files are saved with a .rm extention. However, if you are encoding a Variable Bit Rate (VBR)

RealMedia file, Helix Producer saves the file with a **.rmvb** extension to distinguish it from normal RealMedia files. See "About VBR Encoding" on page 68 to learn more about VBR files.

4. Click **Save**.

Your new destination file is shown in the Output list. If a destination already exists for the job, the new destination will appear on the list below it.

RealMedia File Size

Determining the size of a RealMedia file depends on the choices you made when you choose the encoding settings. As a general rule, if you set Helix Producer to create a better quality output, the size of the RealMedia file will be larger. Settings that can affect file size include:

- number of audiences encoded
- total bit rate of audio and video for each audience

See Chapter 7: Choosing Audiences for an Encoding Job on page 55 for more informations about how to change the above settings.

When a RealMedia file reaches the operating system's file size limit (4 GB on Windows, 2GB on Linux), it will start a new file. The new file will have the same name as the old file, but with a 1 added (for example, **file.rm** and **file1.rm**). You can combine these files together using a SMIL file. Refer to the *RealNetworks Production Guide* for more information.

Broadcasting Live to a Server

You can choose to use a Helix Universal Server as a destination for the encoded output, sending the encoded streaming media directly to a server, and then to your audience as soon as it is being encoded. This method of encoding is called *live broadcasting*.

Live broadcasting
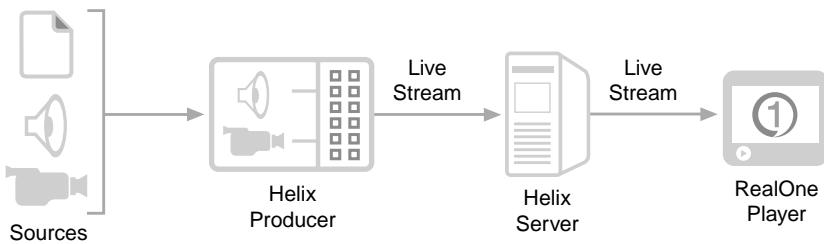
To initiate the live broadcast, you need to tell Helix Producer some information about the server. This information is saved as a Server Destination. You can add more than one Server Destination to a job. When a job has more than one Server Destination defined for it, the encoded media is sent to each server at the same time.

There are two ways to use Server Destinations:

1. Create a Server Destination just for the encoding job.

   If you use this method, you set up a server as your destination, but the settings you use can't be used for other jobs.

2. Use a **Server Template** as a destination.

   If you want to use the Server Destination in future encoding jobs, you can also save it as a Server Template. Using this method, a file is stored on your computer with all the settings you have defined for the Server Destination. See "Using Server Templates" on page 51 for more information.

The rest of this section defines the different broadcasting methods available to you and shows you how to set up a Server Destination for each method. The following is a summary of the methods:

- Push using Account-Based Login
- Push using Password-Only Login
- Multicast Push
- Pull
- Legacy Push (for pre-Helix Universal Servers)

Setting Up an Account-Based Push Broadcast

Account-based Push broadcasting is the standard way of transmitting encoded, streaming media through the server to the audience. It is a the best choice for when you want the reliability of the new Helix features, but also want a simple method to set up.

The main benefit of Push broadcasting is that when your audience connects to the server, there is no delay caused by connecting the producer to the server. The audience receives the data quickly since the server has the information about your broadcast indexed.

The figure below shows you the interaction between Helix Producer, Helix Universal Server, and RealOne Player in an Account-based broadcast.

Account-Based Push broadcast model
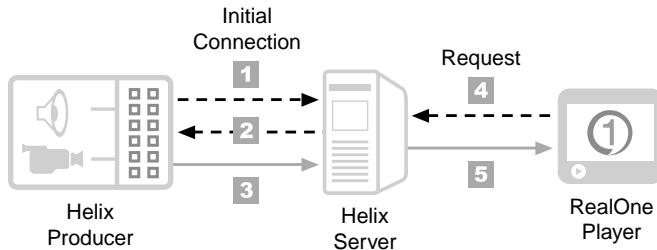
In Step 1, Helix Producer establishes an initial connection with a Helix Universal Server. With this connection, Helix Producer sends a username and password that the server will attempt to authenticate.

In Step 2, once the broadcast is authenticated, the server sends back information to the producer over the same initial connection. This information tells the producer how to make the broadcast connection in the next step.

In Step 3, the broadcast connection is made between Helix Producer and Helix Universal Server. When the broadcast begins, the encoded packets are sent to the server, regardless of whether any players have requested the broadcast.

In Step 4, an audience member uses his or her RealOne Player to request the broadcast from the server.

In Step 5, Helix Universal Server sends the encoded packets that it has been receiving to the audience member's player, which begins decoding and playing the streaming media broadcast.

This type of broadcast is recommended if you do not have much experience setting up live broadcasts with servers, because it requires the minimal amount of information regarding the Helix Universal Server you will be broadcasting to. Get the following information you need to set up the broadcast from your server administrator:

- Server address
- Server port
- User Name
- Password

➤ **To set up a Push, Account-Based Login Server Destination:**

1. From the main window, select **File>Add Server Destination**.

   The Server Destination dialog opens.

2. Enter a **Destination Name**.

   This name will appear in the Output section in the main window. And if you save these settings as a Server Template, it will be the name of the template.

3. Enter a **Stream Name** for the encoded output. This is the name of the file that will be used for the live broadcast.

4. Select **Push, Account-Based Login (Server 9)** from the **Broadcast Method** list.

5. Select the type of connection protocol that you will use. See "Transport Protocols" on page 43 for a description of the protocols available to you.

6. Now enter the other **Broadcast Method Settings** for the Server Destination using the following list for a description of the settings:

   a. **Server Address**
      The IP address or name of the Helix Universal Server used for the broadcast (example: server.real.com).

   b. **Path**
      The directory path on the Helix Universal Server where your broadcast will be accessed by your audience (example: /mydirectory). This setting is optional.

   c. **Port**

The port on the Helix Universal Server that will be sent packets. The default is 80.

d. Username

The name used to authenticate the connection to the server.

e. Password

Your password needed to connect to the server. You can also select to remember your password for future connections to the same server.

7. If you need to set the Advanced Broadcast options (only recommended for advanced users), click the **Advanced Options** button. See "Advanced Broadcast Options" on page 41 for more information on setting these options.

8. (Optional) To save these settings as a Server Template:

a. Click the **Templates** button.

b. Select **Save as Template**.

The Server Destination is now saved as a Server Template that you can use for other encoding jobs.

9. Click **OK** to save the Server Destination.

Advanced Broadcast Options

The following options allow you to define different aspects of the broadcast stream. You can adjust how often the broadcast will attempt to reconnect, what and how often information is shared between Helix Producer and Helix Universal Server, and how the broadcast is protected against lost data.

You use the Advanced Options - Push dialog box to set these options. To open this dialog box, click on the **Advanced Options** button on the Server Destination dialog box.

The following Advanced options can be set:

Attempt to Reconnect

For TCP, you have the option to force Helix Producer to attempt to reconnect to the server if there is a lost connection. You can also specify the amount of time Helix Producer waits between attempts.

Metadata Resend Interval

Helix Producer sends metadata—simple text information about the stream—to the server on a regular basis. The server needs to receive this information before it can properly broadcast the stream.

When you set this value, you determine how often the metadata text will be sent in the stream. Consequently, this value affects how quickly the server can reconnect if the stream is disconnected for any reason.

Statistics Update Interval

Helix Universal Server sends back various statistics to Helix Producer during a broadcast. This option allows you to specify the time between updates.

Listen Address

This is the IP address of your machine where Helix Producer will listen for resend requests from the server. Choose an IP address from the drop-down list.

Allow Helix Universal Server Packet Resend Requests

If this setting is selected, Helix Producer will allow requests from the server to resend lost data packets. Helix Universal Server has a similar option. If the server does not have the option selected, selecting it here will have no effect.

Forward Error Correction Percentage

FEC packets are used to strengthen the stream against data packet loss. They accomplish this by containing redundant information about the preceding packets. This redundant information can be used to reconstruct the stream in the event of packet loss.

The value entered here indicates the percentage of packets that are FEC packets. The greater the percentage, the greater the packet loss that can be handled. If you set this setting to 100%, you are creating a redundant stream to protect against packet loss.

Forward Error Correction Latency

When the FEC percentage is 100%, creating a redundant stream, the FEC latency defines the number of seconds to separate each redundant packet.

Multicast Time to Live

This value indicates the number of router hops that the stream will cross before being blocked. The higher the value, the less you limit the spread of a multicast, but it is less secure.

Transport Protocols

When you set a transport protocol, you are telling Helix Producer how it will communicate with the server. Your choice of protocol determines how data, in the form of packets, will be sent over the connection once broadcasting begins.

There are two options for protocol when connecting to a server: UDP and TCP.

About UDP

UDP is a "connectionless" protocol. Helix Producer, when using UDP, doesn't need to communicate constantly with the server to keep the connection. Only when the server needs a packet of data resent to it, will it request the packets from the producer. This reduces the load on the server that would occur if the server need to send and wait for acknowledgement of packets. Thus, using UDP enhances the performance of your broadcast.

UDP is recommended whenever possible, but TCP (described below) is required if firewalls or other network devices prevent the free transmission of UDP packets. Firewalls are often encountered when encoding over the Internet.

About TCP

TCP protocol is, converse to UDP, a "connection" protocol. When using TCP, a two-way connection exists between Helix Producer and the server. This means that if a packet is lost in transmission, the network requests a packet to be resent.

While TCP is a reliable protocol, the downside is that instead of the server being responsible for lost packets, the network layer is responsible. As a result, a Helix Universal Server loses control of how packets are resent and the timeliness of entering them into the stream. This means that situations occur when a packet is requested by the network and it is resent even if the packet is no longer needed (the stream is past its time to be decoded). This situation results in unnecessary network utilization and reduced efficiency.

TCP is best for use across a firewall where packet loss is greatest so that the stream can be adequately sent to the server.

Setting Up a Password-Only Push Broadcast

This type of broadcast is only recommended if you have experience setting up live broadcasts with servers, and if you are an administrator of the server that

you will use for the broadcast or you are working in coordination with the administrator of that server.

The benefit to using a Password-Only broadcast is that an initial, authenticating connection does not need to be made between Helix Producer and Helix Universal Server before the broadcast can begin. Only a password is needed to authenticate, and that is sent with the broadcast. However, setting up a Password-Only broadcast is difficult without access to the server itself, because with no initial connection, there is no feedback from the server if the broadcast fails. And if it is set up incorrectly, your broadcast will not reach the server.

The figure below shows you the interaction between Helix Producer, Helix Universal Server, and RealOne Player in a Password-Only broadcast.

Password-Only Push broadcast model
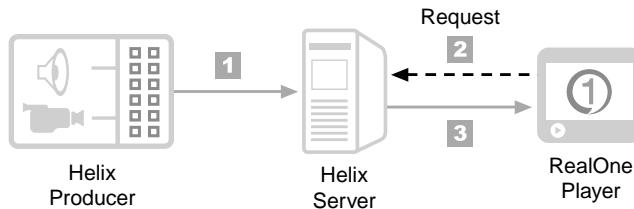
In Step 1, the broadcast connection, using the correct password, is made between Helix Producer and Helix Universal Server. When the broadcast begins, the encoded packets are sent to the server, regardless of whether any players have requested the broadcast.

In Step 2, an audience member uses his or her RealOne Player to request the broadcast from the server.

In Step 3, Helix Universal Server sends the encoded packets that it has been receiving to the audience member's player, which begins decoding and playing the streaming media broadcast.

This type of broadcast is recommended only if you have much experience setting up live broadcasts with servers.

➤ **To set up a Push, Password-Only Login Server Destination:**

1. From the main window, select **File>Add Server Destination**.

   The Server Destination dialog opens.

2. Enter a **Destination Name**.

This name will appear in the Output section in the main window. And if you save these settings as a Server Template, it will be the name of the template.

3. Enter a **Stream Name** for the encoded output. This is the name of the file that will be used for the live broadcast.

4. Select **Push, Password-Only Login (Server 9)** from the **Broadcast Method** list.

5. Select the type of connection protocol that you will use. See "Transport Protocols" on page 43 for a description of the protocols available to you.

6. Now enter the other **Broadcast Method Settings** for the Server Destination using the following list for a description of the settings:

    a. **Server Address**

    The IP address or name of the Helix Universal Server used for the broadcast (example: server.real.com).

    b. **Path**

    The directory path on the Helix Universal Server where your broadcast will be accessed by your audience (example: /mydirectory). This setting is optional.

    c. **Port Range**

    Specifies a range of ports on the server that will be sent packets. Using a range of ports instead of just one minimizes the load on any one socket's memory, thereby keeping the server from dropping packets from any one overloaded port. The default range is 30001-30020.

    d. **Password**

    Your password needed to connect to the server. You can also select to remember your password for future connections to the same server.

7. If you need to set the Advanced Broadcast options (only recommended for advanced users), click the **Advanced Options** button. See "Advanced Broadcast Options" on page 41 for more information on setting these options.

8. (Optional) To save these settings as a Server Template:

    a. Click the **Templates** button.

    b. Select **Save as Template**.

The Server Destination is now saved as a Server Template that you can use for other encoding jobs.

9. Click **OK** to save the Server Destination.

Setting Up a Multicast Push Broadcast

This type of broadcast is only recommended if you have experience setting up live broadcasts with Helix Universal Servers, and if you are an administrator of the server that you will use for the broadcast.

A multicast is designed for a broadcast across a network so that many servers can receive the streaming data. Instead of sending one stream to each server, all servers hook into the same stream. This conserves network bandwidth and server load when you want multiple servers to stream RealMedia content.

Multicast Push model

Helix
Producer
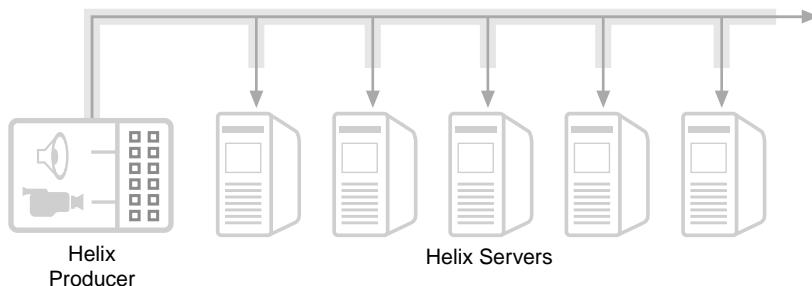
Helix Servers

Without using multicast, Helix Producer would send a separate broadcast stream to each server. So in the above diagram, there would be five different broadcasts.

Multicast broadcasting can also be used for broadcasting across a one-way satellite network where two-way connections are not possible.

➤ **To set up a Push, Multicast Server Destination:**

1. From the main window, select **File>Add Server Destination**.

   The Server Destination dialog opens.

2. Enter a **Destination Name**.

   This name will appear in the Output section in the main window. And if you save these settings as a Server Template, it will be the name of the template.

3. Enter a **Stream Name** for the encoded output. This is the name of the file that will be used for the live broadcast.

4. Select **Push, Multicast (Server 9)** from the **Broadcast Method** list.

5. Enter the **Broadcast Method Settings** for the Server Destination using the following list for a description of the settings:

   a. **Multicast Address**
      The multicast IP address that Helix Producer will broadcast to. This address is a "virtual" IP address that will be used by each server to access the broadcast.

   b. **Path**
      The directory path on the Helix Universal Server where your broadcast will be accessed by your audience (example: /mydirectory). This setting is optional.

   c. **Port Range**
      Specifies a range of ports on the server that will be sent packets. Using a range of ports instead of just one minimizes the load on any one socket's memory, thereby keeping the server from dropping packets from any one overloaded port. The default range is 30001-30020.

   d. **Password**
      Your password needed to connect to the server. You can also select to remember your password for future connections to the same server.

6. If you need to set the Advanced Broadcast options (only recommended for advanced users), click the **Advanced Options** button. See "Advanced Broadcast Options" on page 41 for more information on setting these options.

7. (Optional) To save these settings as a Server Template:

   a. Click the **Templates** button.

   b. Select **Save as Template**.
      The Server Destination is now saved as a Server Template that you can use for other encoding jobs.

8. Click **OK** to save the Server Destination.

Setting Up a Pull Broadcast

Pull broadcasting is different from Push broadcasting in that a connection with a server is not made until the broadcast is requested by a RealOne Player. The broadcast will still be encoded, but it will not be sent to the server until requested.

This method gives you the most control over the broadcast from the server side of the connection. It is best for when your broadcast must always be available but is seldom played, because some time is needed for the first RealOne Player to request the broadcast, the server to relay the request to the producer, and then for the producer to send the broadcast to the server. Another benefit is that bandwidth is saved since no broadcast is sent, unless requested.

The figure below shows you the interaction between Helix Producer, Helix Universal Server, and RealOne Player in a Pull broadcast.

Pull broadcast details
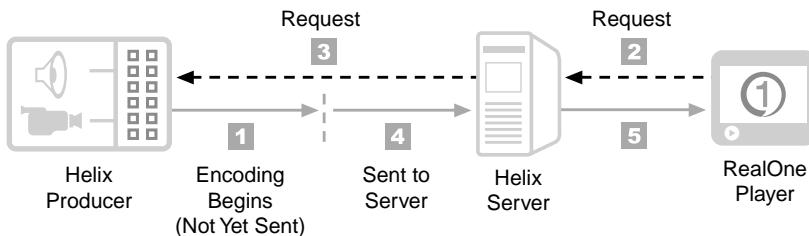
In Step 1, Helix Producer begins encoding live streaming media, but the output is not sent out as a broadcast.

In Step 2, an audience member uses his or her RealOne Player to request the broadcast from a Helix Universal Server.

In Step 3, the server makes a request to the producer that a broadcast should begin. Within this request, the server sends the information needed to start a broadcast. Also, once a broadcast connection is made, the server will continue to send requests that the broadcast is kept alive as long as the player is also requesting the broadcast ("keep alive" requests).

In Step 4, the broadcast connection is made between Helix Producer and Helix Universal Server. The broadcast will continue until the producer, within a specified time, does not receive requests from the server to keep it alive.

In Step 5, Helix Universal Server sends the encoded packets that it has been receiving to the audience member's player, which begins decoding and playing the streaming media broadcast.

➤ To set up a Pull Server Destination:

1. From the main window, select **File>Add Server Destination**.

   The Server Destination dialog opens.

2. Enter a **Destination Name**.

   This name will appear in the Output section in the main window. And if you save these settings as a Server Template, it will be the name of the template.

3. Enter a **Stream Name** for the encoded output. This is the name of the file that will be used for the live broadcast.

4. Select **Pull (Server 9)** from the **Broadcast Method** list.

5. Now enter the **Server Settings** for the Server Destination using the following list for a description of the settings:

   a. **Local IP Address**
      The IP address on the Helix Producer's computer where a server can look for your encoded stream. Choose an address from the drop-down list.

   b. **Path**
      The directory path on the Helix Universal Server where your broadcast will be accessed by your audience (example: /mydirectory). This setting is optional.

   c. **Producer Listening Port**
      The port located on the Helix Producer's computer where a server can look for your encoded stream.

   d. **Password**
      Password used to authorize connections from the server. You can also select to save your password.

6. If you need to set the following option (only recommended for advanced users), click the **Advanced Options** button:

   • **Helix Universal Server Connection Timeout**

If you select this setting, you are allowing a server to try to connect again after a connection fails. You can set how long the server must wait before the attempted connection is aborted.

7. Click **OK** to close the Advance Options - Push dialog.

8. (Optional) To save these settings as a Server Template:

   a. Click the **Templates** button.

   b. Select **Save as Template**.

   The Server Destination is now saved as a Server Template that you can use for other encoding jobs.

9. Click **OK** to save the Server Destination.

Setting up a Legacy Push Broadcast

Before you can connect to a legacy version of Helix Universal Server (before version 9), you will need to know some vital information about the server. Within the procedure below is a list of the primary information about the server that you will need in order to set up a simple broadcast. You can get the appropriate information from your server administrator or from your Internet Service Provider if they are providing access to a server. Or you can have your administrator create a Server Template for you. See "Using Server Templates" on page 51 for more information.

➤ To set up a Legacy Push Server destination:

1. From the main window, select **File>Add Server Destination**.

   The Server Destination dialog opens.

2. Enter a **Destination Name**.

   This name will appear on the Output list in the main window. And if you save these settings as a Server Template, it will be the name of the template.

3. Enter a **Stream Name** for the encoded output.

4. Select **Legacy Push (8.x, 7.x, G2)** from the **Broadcast Method** list.

5. Now enter the **Broadcast Method Settings** for the Server Destination using the following list for a description of the settings:

   a. **Server Address**

The IP address or domain of the Helix Universal Server used for the broadcast (example: server.mycompany.com).

b. Path

The directory path on the Helix Universal Server where your broadcast will be accessed by your audience (example: /mydirectory). This setting is optional.

c. Port

The port number that the Helix Universal Server uses to accept connections for broadcasted streams (example: 4040).

d. **Usernam**e

Your log-in name used to connect to the server. This log-in name is created by a server administrator.

e. Password

Your password needed to connect to the server. You can also select to save your password for future connections to the same server.

f. Select the type of connection protocol that you will use. See "Transport Protocols" on page 43 for a description of the protocols available to you.

6. (Optional) To save these settings as a Server Template:

a. Click the **Templates** button.

b. Select **Save as Template**.

The Server Destination is now saved as a Server Template that you can use for other encoding jobs.

7. Click **OK** to save the Server Destination.

Using Server Templates

You can save a Server Destination as a Server Template. Creating a template gives you the ability to use a Server Destination for later jobs. You can also share the template with others who will be using the same server for their encoding jobs.

By creating a Server Template, you are creating a text file that contains information about the server. When you subsequently use the template as a destination for a job, that information is copied into the job.

➤ To create a Server Template:

1. From the Output section of the main window, select a Server Destination that you have created for your encoding job.

2. Click the **Edit** button.

   The Server Destination dialog opens.

3. Click the **Templates** button.

   A menu opens listing the Server Templates that you have created.

4. Click on **Save As Template**.

   The Server Destination is now saved as a Server Template. The next time you click on the Templates button, the template will appear in the list.

When you have Server Templates saved on your computer, you can use those templates to define a destination in your job. This gives you a quick way to set up a Server Destination.

➤ To use a Server Template as a destination:

1. From the main window, select **File>Add Server Destination**.

   The Server Destination dialog opens.

2. Click the **Templates** button.

   All of the Server Templates that you have created are listed.

3. Select a template from the list.

   The settings associated with the template appear automatically in the Server Destination dialog.

4. Click **OK** to return to the main window.

Editing Server Templates

All Server Templates are editable. Thus, you don't have to completely recreate a template if you only need to make a small change to it.

You edit a template using the Server Templates dialog. This dialog also shows you all Server Templates that have been saved to your machine. It allows you to quickly scan through the different templates so that you can edit them or review their settings.

When you make changes to a Server Template, you are only making the changes to the information saved in the template file. Thus, the changes will

not take place in any jobs that were previously created using the template. To add these changes to a job, you must remove the destination from the output list in that job, then add a new destination using the revised template.

➤ To edit a Server Template:

1. From the main window, select **Edit>Server Templates** from the main menu.

   The Server Templates dialog opens, giving you a list of all Server Templates saved on your computer.

2. Select a template from the list.

3. The template's settings appear in the right side of the dialog.

4. You can now edit the settings for the template, using the procedures described in the above sections.

5. Click **Apply** to save the changes to the template.

You can also create a new template by clicking the Duplicate button. This way, an entirely new template name will appear in the list of templates, and a new Server Template file is saved on your computer.

Adding Clip Information

Adding clip information allows you to attach pertinent information about your streaming content, such as the title, who created it, and a few keywords, that describes your clip to your potential audience. This information is used to help your audience find out more about your content.

➤ To add information about the output clip:

1. In the Job Manager, select a job.

2. From the Helix Producer main window, select **Settings>Show Clip Information.**

   The Clip Information palette opens.

3. Enter the following information in the corresponding areas:

   a. **Title**: enter the name of the RealMedia clip that you want to appear in your audience's RealPlayers.

   b. **Author**: enter the name of the person who created the clip.

c. **Copyright**: enter the year and owner of the clip (for example, "copyright 2002, ABC Corporation").

d. **Keywords**: one or two words that will help your audience search for your clip.

e. **Description**: type in a brief summary of what your clip is all about.

f. **Rating**: choose a rating for your clip based on its contents.

CHOOSING AUDIENCES FOR AN ENCODING JOB

This chapter describes how you can set up Helix Producer to encode streaming media tailored to your audience. You will learn how to choose the type of audio and video you are encoding to create a better output. You will learn how to select your audiences, either by using a preset audience or create a new one and save it as a template. Finally, you will learn about Variable Bit Rate audiences.

Overview: Audiences

When you are setting up a job, you must choose one or more audiences that you will be targeting. These audiences will be encoded as separate streams, and each will be encoded with settings based on the bit rate for the audience.

For a summary of how audiences work within Helix Producer, see "Targeting Audiences" on page 10.

The basic steps to selecting audiences is as follows:

1. Change the settings that determine how every audience will encode streaming media.

2. Choose the audience or audiences that you want to encode for.

3. Edit any audience to reflect your encoding needs.

4. Save the modified audience as a template. This step is optional.

Each of the above steps are covered in more detail throughout this chapter.

Adjusting Encoding Settings for All Audiences

You can change various settings to reflect how you want Helix Producer to encode the final streaming media. These settings allow you to adjust the output depending on a number of factors:

- type of audio (if any) that is encoded

- style of the video (if any) that is encoded

- video codec that you want to use to encode the video

- whether to use a high-quality resampler when encoding the audio

- whether to use 2-pass encoding

- whether the video is resized

These encoding settings will affect all destinations that you have defined in the output. Each setting can be found in the Encoding Settings section of the Audiences palette.

To open the Audiences palette, click the **Audiences** button on the main window.

> **Tip:** Any changes you make to the Audiences palette will be automatically updated for the current job. You can leave the palette open as you continue to set up a job or switch to another job.

Setting the Audio Mode

When you select the Audio Mode, you are telling the Helix Producer the type of audio being encoded. You can choose between **Voice** and **Music** for the Audio Mode.

Separate audio codecs exist that encode either voice data or music data better. Each audience that you will select has different codecs assigned for either Voice Audio Mode or Music Audio Mode. Thus, this setting allows you to specify that either a Voice codec or a Music codec is used when you are encoding your audio.

If you are unsure of which Audio Mode to choose because your audio content is mixed, the Music codecs provide the greater range and work best in mixed-use cases. Also, if you are encoding at a high bit rate, the Music codecs can give you a better quality, even for audio that contains all spoken word.

Setting the Video Mode

The Video Mode you select in the Audiences palette should determined by the amount of motion within the video. Examples of video clips with lots of "motion" are shots of people running, or quick pans of the camera from one

person to another and back again, with each person talking in an animated fashion. Alternately, examples of video without motion are simple shots without any panning, or shots of scenery.

You can choose from the following:

- For clips with a wide range of motion, the best setting is **Normal Motion**.

  If you desire better clarity during scenes with much motion, you can choose the **Sharpest Image** mode. This mode will use more encoding time on the clarity of the image, but at the expense of frame rate.

- If you would rather have more continuity during high-motion scenes, you can choose **Smoothest Motion**. More encoding time will be spent on maintaining the frame rate, but will be at the expense of each frame's image clarity.

- You can also choose to completely focus on image clarity by choosing the **Slide Show** mode. The output appears as a series of still photos.

Choosing a RealVideo Codec

In the Audiences palette, you can choose which version of RealVideo you will use to create your streaming media clips. Each version uses a different method to compress your original video data into a format that is best for streaming.

A new video codec, RealVideo 9, is now included in Helix Producer. RealVideo 9 allows you to create the best quality video for any bit rate, especially at high-bandwidth rates. The following is a summary of the new features of RealVideo 9:

- 30% improvement in video quality over RealVideo 8 and a 50% improvement over RealVideo G2.

- ½ screen video at dial-up rates

- VHS quality at everyday broadband rates (starting at 160 kbps)

- Near DVD at 500 kbps for download or streaming on high speed networks

Choosing a RealVideo codec affects who can view your streaming video clip. Only those members of your audience who have a player with the selected video codec installed can play back the video. While the newest video codec gives you the best video, if you know that many people in your target audience have an older player version, it may be best to select an older codec.

Using the High Quality Audio Resampler

When the audio sample rate of the source audio is different than required from the audio codec, Helix Producer must resample the audio. If you choose this option, the resampled audio will be of a higher quality, but the encode will need more resources (more time for encoding on-demand, and more processing power for live encoding).

Using 2-Pass Encoding

In the Audiences palette, you can also select whether or not you want Helix Producer to use 2-Pass Encoding every time that it encodes RealMedia for the job. This setting will only be in effect when you are encoding files as your input to a RealMedia file output (i.e., it is not applied for live capture or live broadcasts). If you define your input and output (even one of multiple destinations) as anything other than files, this setting will be ignored.

As its name indicates, each time a job is encoded with 2-Pass Encoding enabled, the encoding process will be comprised of two passes. The first pass analyzes the data in the input media file to determine where the most bits are required in the file. The second pass encodes the file using the analysis provided by the first pass.

Using this feature results in better video for your RealMedia file, but the drawback is that the encoding process will take longer than encoding in a single pass.

Resizing the Video

If you want the video of the encoded output to be resized, you can make that adjustment in the Audiences palette. Enter a value (in pixels) for the new height and width. If you leave these settings blank, the size of the original video will be used for the encoded output video. If you resize the video to a much larger size, the output video will not have the same quality.

If you want to keep the original video's ratio of height to width in the output video, select **Maintain Aspect Ratio**. If you do not keep the original ratio, the output video image will appear stretched or squashed.

If you are also cropping the input video, the crop will occur before resizing. See "Cropping the Input Video" on page 31 for more information on cropping video.

Using the High Quality Resizing Filter

Whenever you resize input video, you can degrade the quality of the image. The High Quality Resizing filter is designed to make up for this degradation if it is selected for resized video. The trade-off is that more processing, and time, is needed during the encode to fix the resized video.

➤ **To turn on the High Quality Resizing filter:**

1. From the main window, click the **Video Filters** button.

   The Video Filters palette opens.

2. Select the checkbox labelled **High Quality Resize**.

   When selected, the filter is enabled.

Choosing Audiences

Before you encode, you choose the audience or audiences you want to encode for. For each audience, a separate stream is encoded based on the type or speed of Internet (or Intranet) connection that your audience members possess. For example, if you predict that your audience will be primarily DSL or Cable Modem users, you will want a stream tailored for that bit rate.

When you choose to encode for an audience, you are telling Helix Producer to create a stream for that audience. You can choose more than one audience for an encoding job, but the size of the encoded output will grow with every stream added.

Helix Producer comes with a variety of audience definitions that have been created for a variety of audience types. These installed audience definitions are Audience Templates. An Audience Template is a text file that contains information about the settings for the audience. When you subsequently use a template to define an audience for a job, that information is copied into the job.

While you can change the settings in an Audience Template, it is recommended that you become familiar with these default Audience Templates before you start changing their settings. See "Creating New Audience Templates" on page 66 for more information on creating your own templates.

Below is a list of the default Audience Templates that are installed with Helix Producer and their respective bit rates.

12 kbps Substream for 28k Dial-up	12 kbps
16 kbps Substream for 28k Dial-up	16 kbps
28k Dial-up	20 kbps
Wireless	20 kbps
26 kbps Substream for 56k Dial-up	26 kbps
56k Dial-up	34 kbps
64k Single ISDN	45 kbps
128k Dual ISDN	80 kbps
150k LAN (Local Area Network)	150 kbps
256k DSL or Cable modem	225 kbps
384k DSL or Cable modem	350 kbps
512k DSL or Cable modem	450 kbps
768k DSL or Cable modem	700 kbps
Download - VHS quality	750 kbps
Download - DVD quality	1500 kbps

➤ **To use an Audience Template to define an audience for your encoding job:**

1. Open the Audiences dialog.

2. Select an Audience Template from the list of Templates.

3. Click the right arrow button. ->

   A copy of the settings is the selected Audience Template is transferred into the job, and the audience name appears in the list of audiences used for the job.

4. Use the templates to copy as many audiences as you need to your job.

While you can add a number of audiences, each one that is added adds more encoding time and uses computing resources during the encode. It is recommended to use two or three different ones that you know you need.

You can also remove an audience that has been added to a job.

➤ To remove an audience from your encoding job:

1. Open the Audiences dialog.

2. In the list of Audiences In Job, select the audience that you want to remove.

3. Click the **Delete** button to delete the audience from the list.

Editing Audiences Selected for a Job

While the Audience Templates that are installed with Helix Producer are created specifically for the target bit rate, you can fine-tune these settings once you have copied the template into your encoding job.

When you edit a job's audience, the changes you make will only apply to the current job—and not to the Audience Template itself. See "Creating New Audience Templates" on page 66 for information on editing a template.

This section describes the different settings that make up an audience and shows you how to change them to suit your needs.

Adjusting Video Stream Settings for an Audience

One component of an audience is the video stream. The way in which Helix Producer creates the stream for the audience is based on a variety of settings that you can adjust.

Encoding a video stream is a complex task, so there are a variety of settings that you can use to adjust how the video codec creates the output video stream. You can set a target for the bit rate, set a target frame rate, and other advanced options.

Setting the Bit Rate for a Video Stream

The bit rate for an audience's video stream is the rate at which encoded information (bits) will be sent to your intended audience. You can adjust this to any number you wish, but it should be realistic. A 56 kbps modem won't be able to receive 100 kbps. It won't even be able to receive at 56 kbps because of packet loss during transmission.

When an audience template uses a constant bit rate, that simply means that at all times during playback, an audience will stream at one bit rate. The advantage to using a constant bit rate is that you can maintain a constant bit rate for connections that require a hard and fast limit on the bit rate. The

disadvantage is that video clips with variations in movement—or other factors that can affect the data that needs to be streamed—will lose video quality during those times where lots of data needed to be sent.

> **Note:** For information on using Variable Bit Rate (VBR) encoding, see "Using Variable Bit Rate (VBR) Audience Templates" on page 68.

➤ **To set a constant bit rate for an audience:**

1. From the Audiences palette, select a non-VBR audience from the list of audiences that you have selected to use with your job.

2. Click the **Edit** button.

   The Audience Properties dialog for that audience opens.

3. Enter a value for the **bit rate** in the **Constant Target Bit Rate** field.

   You should not enter a value greater than the bit rate you have available for the video stream.

   > **Tip:** To calculate the maximum bit rate for video available, add the bit rate of the selected Music Codec to the **Average video bit rate with music** that is listed in the Audio/Video Encoding section of the dialog.

4. Click **OK** to save the settings to this audience.

Adjusting the Target Frame Rate

The frame rate is the frequency at which the video stream is updated with new frames. This value is measured in frames per second. A high frame rate gives a smoother appearance to the motion of the video because more frames are sent per second to the audience's players. A lower frame rate can give a choppy appearance to the video. And a very low frame rate will make the video appear like a number of still images.

Some examples of common frame rates are: 24 frames per second (fps) is standard for film, whereas 30 fps is the standard for NTSC broadcast video. For lower bit rates, a frame rate of 10-15 fps is generally the best compromise of motion and quality.

➤ **To set the target frame rate for an audience:**

1. From the Audiences palette, select an audience from the list of audiences that you have selected to use with your job.

2. Click the **Edit** button.

   The Audience Properties dialog for that audience opens.

3. In the Target Frame Rate field, enter a number from 0.01 to 30.

4. Click **OK** to save the settings to this audience.

Adjusting Advanced Video Stream Options

In addition to the above, basic settings for the video stream, you can adjust more advanced settings that give you more control about how the video stream is created.

Setting the Maximum Startup Latency

This settings allows you to set the maximum time the clip will pre-buffer before playing begins. The pre-buffer time is used by the video codec to store bits that will be necessary during high-action scenes. The larger the value, the longer the user before the clip starts, but the video's quality will be better.

➤ To set the startup latency:

1. From the Audiences palette, select an audience from the list of audiences in your job.

2. Click the **Edit** button.

   The Audience Properties dialog for that audience opens.

3. Click the Advanced Video Options button.

   The Advanced Video Options dialog opens.

4. Enter a number for the **Maximum Startup Latency** between 4 and 25 seconds.

5. Click **OK** to save the changes.

Setting the Maximum Time Between Keyframes

This setting allows you to choose how often keyframes are encoded. Keyframes are frames of video that are encoded in their entirety. In between these keyframes, each frame of video is encoded as a change from the earlier keyframe. Thus, less information about the video needs to be encoded.

Typically, the RealVideo codec will put a keyframe when needed (the scene changes drastically), but will put one every ten seconds if not needed. If you add more keyframes, more information will be encoded resulting in a greater bit rate required to stream the clip.

➤ To set the time between keyframes:

1. From the Audiences palette, select an audience from the list of audiences in your job.

2. Click the **Edit** button.

   The Audience Properties dialog for that audience opens.

3. Click the Advanced Video Options button.

   The Advanced Video Options dialog opens.

4. Enter a number for the **Maximum Time Between Keyframes**, a value between 0 and 60 seconds.

5. Click **OK** to save the changes.

Enabling Loss Protection

You also have an option to add error correction information to the encoded file that will harden the file against packet loss when streaming. These lost packets may be reconstructed by the player.

➤ To enable loss protection:

1. From the Audiences palette, select an audience from the list of audiences in your job.

2. Click the **Edit** button.

   The Audience Settings dialog for that audience opens.

3. Click the Advanced Video Options button.

   The Advanced Video Options dialog opens.

4. Select whether you want to use **Loss Protection**.

5. Click **OK** to save the changes.

Adjusting Audio Stream Settings for an Audience

All streaming audio clips use a RealAudio "codec." Codec is a short name for *compressor/decompressor algorithm*. Helix Producer uses RealAudio codecs to tell a computer how to convert audio input into streaming audio clips. On the receiving end, decoding software, such as RealOne Player, use these same codecs to expand the streaming clips into digitized audio data.

RealNetworks has developed an array of RealAudio codecs, and each one is designed for a type of audio encoded at a specific bit rate. For example, one

RealAudio codec compresses mono music for 32 kbps. Another compresses stereo music for that same bit rate. For a detailed list of RealAudio codecs, see "RealAudio Codec Reference" on page 113.

RealAudio codecs squeeze down the audio data from the source by, in part, throwing out data, which makes RealAudio a "lossy" compression format. RealAudio doesn't delete data indiscriminately, though. It first jettisons portions you can't hear, such as very high and very low frequencies. Next, it removes as much data as needed while keeping certain frequencies intact.

The type of codec you choose (voice, mono, or stereo) determines which frequencies stay and which go. Voice encoding favors frequencies in the normal human speaking range. Music (mono and stereo) encoding retains a broader frequency range.

Changing RealAudio Codecs for Audio Only Encoding

There are two types of encoding that you can choose RealAudio codecs for: Audio Only Encoding and Audio/Video Encoding. Audio Only codecs are used when the input for a job has no video data—just audio.

➤ **To change an audience's RealAudio codec used during audio-only encoding:**

1. From the Audiences palette, select an audience from the list of audiences that you have selected to use with your job.

2. Click the edit button.

   The Audience Settings dialog for that audience opens.

3. In the Audio Only Encoding section, choose a RealAudio codec that you will use for the **Voice Codec**.

   This codec will be used for this audience when you choose Voice as the Audio Mode in the Encoding Settings dialog.

4. Choose a RealAudio codec that you will use for the **Music Codec**.

   This codec will be used for this audience when you choose Music as the Audio Mode in the Encoding Settings dialog.

5. Click **OK** to save the setting to the audience.

Changing RealAudio Codecs for Audio/Video Encoding

Audio/Video Encoding codecs are for when you are also encoding video data. Thus, you have less bandwidth to work with and need to select a lower bit rate for the codec.

➤ To change an audience's RealAudio codec used during audio/video encoding:

1. From the Audiences palette, select an audience from the list of audiences that you have selected to use with your job.

2. Click the edit button.

   The Audience Settings dialog for that audience opens.

3. In the Audio/Video Encoding section, choose a RealAudio codec that you will use for the **Voice Codec**.

   This codec will be used when you choose Voice as the Audio Mode in the Encoding Settings dialog.

4. Choose a RealAudio codec that you will use for the **Music Codec**.

   This codec will be used when you choose Music as the Audio Mode in the Encoding Settings dialog.

5. Click **OK** to save the setting to the audience.

   > **Tip:** When you choose an audio codec for Audio/Video encoding, the resulting Video Bit Rate that can be used for the video portion of the encoding job is calculated and displayed for you.

Creating New Audience Templates

Creating an Audience Template gives you the ability to use your audience settings for later jobs. You can also share your Audience Template with others who will be encoding for the same audience.

While you can simply edit one of the default Audience Templates to create your own template, the best way to customize a template is to create a new Audience Template.

➤ To create a new Audience Template:

1. From the main window, select **Edit>Audience Templates.**

   The Audience Templates dialog opens.

2. Select an Audience Template that is most similar to the template that you want to create.

3. Click the Duplicate button.

A copy of the Audience Template is made, and it is added to the template list.

4. Type in a new **Template Name**.

5. You can now edit the settings for the new template, using the procedures described in the above sections.

6. Click the **Apply** button when finished.

   Your new, custom template appears in the list of Audience Templates. And a new template file is saved on your computer so that it can be used when creating encoding jobs.

The rest of this section shows you how to adjust settings for an Audience Template.

Editing Audience Templates

Editing an Audience Template is similar to adjusting settings for an audience. The difference is that the changes you make to a template can be used for future jobs.

You edit a template using the Server Templates dialog. This dialog also shows you all Server Templates that have been saved to your machine. It allows you to quickly scan through the different templates so that you can edit them or review their settings.

When you make changes to an Audience Template, you are only making the changes to the information saved in the template file. Thus, the changes will not take place in any jobs that were already created using the template. To add these changes to a job, you must remove the audience from the Audiences In Job list, then select the template again to copy its settings into the job.

➤ To edit an Audience Template:

1. From the main window, select **Edit>Audience Templates**.

   The Audience Templates dialog opens, giving you a list of all Audience Templates saved on your computer.

2. Select a template from the list.

3. The template's name and settings appear in the dialog.

4. You can now edit the settings for the template, using the procedures described in the above sections.

5. Click **Apply** to save the changes to the template.

> **Tip:** If you have altered and saved the original default templates, you can restore the original templates that are included with Helix Producer by clicking the **Restore Selected Template** button or the **Restore All Templates** button on the Audience Templates dialog.

Using Variable Bit Rate (VBR) Audience Templates

For most video clips that you want to encode, there exists a balance of needs. On one hand, you want to preserve the quality of the encoded video during the high-action scenes. But on the other hand, you need to keep the encode bit rate at a level at which your audience can view the video without interruption by buffering.

Variable Bit Rate (VBR) encoding gives you the ability to balance these needs when you encode with an Audience Template.

About VBR Encoding

Most video clips have different scenes with various levels of complexity. Complexity can arise when there is a large amount of motion in the video, such as a car chase where there are many subjects moving in different directions. Naturally, complex video scenes are the most difficult for Helix Producer to encode.

When you set Helix Producer to encode at a constant bit rate (CBR), all scenes, regardless of their complexity, are encoded equally. Thus, highly complex scenes lose video quality once encoded.

With VBR, however, these complex scenes are treated differently. When you specify a target bit rate for encoding, it is usually below the actual bit rate for the connection. For example, a 384 kbps DSL/ Cable modem audience will be targeted at 350 kbps. This extra bandwidth is generally saved for network congestion or similar constraints on the transfer of the encoded data. If you were to specify a CBR near the threshold bit rate, your audience would lose the connection or suffer rebuffering often. VBR allows Helix Producer to use this extra bandwidth—as needed—to handle the extra bits encoded during complex video scenes.

The following are limitations when using VBR:

- You must use Helix Universal Server 9 for streaming VBR content.

- You can only use one audience for a job if the audience is a VBR template.

Types of VBR Encoding

There are two ways to use VBR encoding. The first option is to specify an average bit rate at which Helix Producer will encode the streaming video clip. The second option is to specify a video quality at which Helix Producer will attempt to encode the streaming video clip.

The table below gives you a summary of the best uses for all three types of encoding possible with Helix Producer: CBR, Average Bit Rate VBR, and Video Quality VBR.

Best Uses for Encoding Types

Types of Encoding	Best Uses
CBR	Narrowband streaming
VBR - Average Bit Rate	Broadband streaming
VBR - Video Quality	Downloadable files, DVD content

Both of the VBR options are defined in detail below.

> **Note:** When you create a VBR RealMedia file, Helix Producer saves the file with a **.rmvb** extension to distinguish it from normal RealMedia files, which have a **.rm** extension.

Average Bit Rate VBR Encoding

This type of VBR encoding tells Helix Producer to try and stay at a target bit rate, on average. The target average bit rate that you specify should be below the maximum bit rate possible for the connection type. In this respect, it is similar to CBR encoding as it stays below the bit rate threshold. Thus, your audience is less likely to lose a connection or experience rebuffering.

But when the video clip becomes more complex, Helix Producer can use the extra bit rate for a limited time so that the video quality can be maintained. Once the complex video scene is over, the specified average bit rate is resumed. If a complex scene is especially long, Helix Producer will lower the bit rate (and the video quality) to keep to the specified average.

Therefore, the best use for average bit rate VBR encoding is for a broadband audience, where the connection can withstand the occasional increase in bit rate.

Video Quality VBR Encoding

This type of VBR encoding is for when you are more concerned about preserving the quality of the video image than losing your audience. Here, you specify a percentage of quality that you want Helix Producer to maintain.

To keep the video quality, the encoded bit rate may remain at a high bit rate—even staying at the maximum bit rate specified—for as long as the video complexity also remains high.

Therefore, the best use for video quality VBR encoding is when you are creating a RealMedia file that will be downloaded by your audience. It is not suited for streaming with a Helix Universal Server.

Creating a VBR Audience Template

Helix Producer comes with a number of templates created with different audiences in mind. Their default settings are suggested settings for a particular audience, allowing you to edit the template to suit your needs. You can edit an existing VBR template, or you can create a new VBR template.

While you can create a VBR template for any bit rate, it is recommended that you only use VBR for bit rates above 150k.

➤ To create a VBR Audience Template:

1. Open the Audience Templates dialog.

2. Select an audience template from the list on the left.

3. In the Audio/Video Encoding section of the dialog, select **Variable Bit Rate**.

4. You can select one of two ways to vary the bit rate:

   a. Select **Target Bit Rate** and enter a value in the field.

      If you use this method, Helix Producer will create a RealMedia clip that uses the target bit rate as the average for the duration of the clip.

   b. Select Target Video Quality and enter a percentage in the field.

If you use this method, Helix Producer will create a RealMedia clip that uses the video quality as a basis for varying data transfer, and will stream at this quality, regardless of bit rate.

5. Enter a value for the **Maximum Bit Rate** that will occur during spikes of data transfer. You cannot enter a value greater than the bit rate you have available for the video stream.

6. Click **Apply** to save the settings to the template.

STARTING AND MONITORING AN ENCODING JOB

This chapter shows you how to start an encoding job, and it describes the different methods that are available to you to monitor a job while it is encoding. You will learn about different encoding statistics presented to you during an encode and tips on how to use those statistics. You will also learn how to use logging features that can be monitored while a job is encoding.

Starting an Encoding Job

If you have created an encoding job, specified what source will be used for the input, and specified the different destinations for the output, all that is left is to start the encode.

The encoding process can be different depending on the type of destination you are encoding to. The following procedure describes what you can expect during an encode for each type of destination.

➤ **To start encoding a job:**

1. Select the job you want to encode from the Job Manager list.

2. Prepare your sources that will be used as the input for the encoding job.

   • If encoding from a file, make sure the file still exists and is in the proper location.

   • If encoding from a live source, make sure your equipment is functioning properly and is connected to your computer's capture card.

3. Click the **Encode** button.

   If there is audio for your encoding job, you will see the Output VU Meter light up.

> **Tip:** Watch the Output VU Meters for readings that reach the top of the scale consistently. This means that your audio is being "clipped" and will have poor sound when your user plays the file back. See "Monitoring and Adjusting Audio" below for more information.

If there is video for your encoding job, you will see the input video (what the source video looks like) in the Input Video Preview window. And you will see the output video (what the encoded video looks like) in the Output Video Preview window.

> **Tip:** If you are encoding for more than one audience, you can choose what audience's video is shown in the Output Video Preview window during encoding. Select the **Audience** from the list above the window.

4. When you are finished encoding, click the **Stop** button. Or, if you have set a duration for the encode, you can wait until the duration is complete.

Monitoring and Adjusting Audio

While you record RealMedia clips, you can monitor the audio levels for both the input audio and the encoded output audio to be sure you are encoding the optimal dynamic range. You monitor audio using the Input audio meter and the Output audio meter, found on the main window next to the video viewers.

On the meters, green indicates a normal reading. Red warns that you are close to an over-modulated input. When the audio is above the possible range, the clipping indicator above the meter lights up. This means that the audio is "clipped."

You can adjust the output of the audio using the Gain palette. The Gain palette allows you to reduce or increase the audio gain for the output audio according to a range that you specify.

➤ To adjust the audio gain:

1. From the main window, select **Settings>Show Audio Gain Control** from the main menu.

   The Gain palette opens.

2. Set the audio gain. You can set it to increase up to +12 dB, or reduce to -12 dB.

Disabling Monitors While Encoding

Disabling the Input video screen, the Output video screen, the Input audio meter, and the Output audio meter can lower the amount of processing power needed during encoding. Consequently, it can increase the frame rate of your encoded file and decrease the time it takes to encode your clip.

➤ To disable viewers during recording:

1. Open the **View** menu from the main Helix Producer window.

   Viewers that are currently active are designated with a checkmark:

   - Input Audio Meter
   - Input Video
   - Output Audio Meter
   - Output Video

2. Deselect a viewer to disable it.

Adjusting File Location Preferences

When you are using Helix Producer, you can adjust different preferences that determine how and where support files are stored. These preferences will be in effect for all jobs.

Changing the Default File Destination Location

You can change the default setting for where Helix Producer creates the output files.

➤ To change the default file destination location:

1. From the main window, select **Edit>Preferences**.

   The Preferences dialog opens.

2. Select the File Locations category.

3. Choose to **Automatically create a file for each job**.

4. Select one of the following options:

   a. **Same Location as Source File**—saves the file in the same directory as the media file used as the source.

   b. **Other**—allows you to specify the full path of the file directory.

5. Click **OK** to save the preferences.

Changing Template Directories

You can change the default directories where your templates will be stored.

➤ **To change the default template directories:**

1. From the main window, select **Edit>Preferences**.

   The Preferences dialog opens.

2. Select the File Locations category.

3. In the Audience Template field, either type in the full path name or click the Browse button to find the directory you wish to use.

4. In the Server Template field, either type in the full path name or click the Browse button to find the directory you wish to use.

5. Click **OK** to save the preferences.

Changing the Temporary Storage Directory

Helix Producer uses a temporary directory while encoding to save data that it needs to complete the encode. You can adjust where this directory is located.

➤ **To change the temporary storage directory:**

1. From the main window, select **Edit>Preferences**.

   The Preferences dialog opens.

2. Select the File Locations category.

3. Select one of the following options in the Temporary Storage Directory section:

   a. **Same Location as File Destination**—saves the temporary data in the same directory as the destination file.

   b. **Other**—allows you to specify the full path of the temporary directory.

4. Click **OK** to save the preferences.

Monitoring Statistics

Helix Producer gives you a variety of statistics about your encoding job that you can monitor during an encode. These statistics give you an overview of the

encoding job, showing you what your settings are for the job and how well the actual encode meets your target settings.

➤ **To open the Statistics window:**

1. From the main window, select the **View** menu item.

2. Select **Show Statistics**.

   The Statistics palette opens.

In the Statistics palette, you are given a variety of statistics that allow you to monitor different aspects of each encoded audience stream as it is encoded. Once the encoding is complete, you can also use the Statistics palette to give you a summary of how the audience streams are encoded. The following statistics appear in the Statistics palette:

Summary of Statistics

Statistic	Description
Audience	The name of the audience stream being encoded.
Total Bit Rate	The total bit rate of the encoded audience stream, in the form of kbps with single decimal precision (or Mbps if the value is over 1 Mbps, with two decimal precision).
Video Bit Rate	The bit rate of the encoded video portion of the audience stream. If there is no video included in an output, this will display n/a.
Audio Bit Rate	The bit rate of the encoded audio portion of the audience stream. If there is no audio included in an output, this will display n/a.
FPS	The current frame rate of the video, in frames per second. After encoding, lists the average frame rate.
Min. FPS	Lists the minimum frame rate, in frames per second, of encoded video that was recorded during encoding.

(Table Page 1 of 2)

Summary of Statistics (continued)

Statistic	Description
Quality	The quality level of the encoded video compared to the source video frames after filtering has been applied, where 100% represents equivalent quality to the source frames, and anything less represents a percentage of the possible quality. After encoding, lists the average quality percentage.
Min. Quality	Lists the minimum quality level of the encoded video that was recorded during encoding.
Pre-Roll	Number of seconds of pre-roll (buffering) that the user will experience when playing your encoded clip.

(Table Page 2 of 2)

These statistics exist in three different states, depending on where you are in the encoding process:

Configuration Statistics

In this state, the Statistics palette shows you statistics for how the streams are configured to be encoded. The bit rates and FPS, as defined in the audiences, are listed.

Encoding Statistics

In this state, the Statistics palette shows you the statistics during encoding. The different statistics will update in real-time, giving you direct feedback on how the job is being encoded. Statistics will not be updated during the analyzing phase of 2-pass encoding.

Summary Statistics

In this state, the Statistics palette shows you the actual statistics for the entire encoding job. Some statistics, such as FPS are listed as averages.

Using Logging Features

When you use Helix Producer, the program creates logs of various tasks that occur. Similar to using statistics, you can use these logs to see if an encode functioned as desired.

There are two ways to access the logs that are created. You can use the Log Viewer or have Helix Producer create a Log File.

Using the Log Viewer

The Log Viewer gives you a real-time view of the log messages as they are being created when using Helix Producer. You also have the ability to filter out types of information during encoding.

➤ To open and use the Log Viewer:

1. From the main window, select the **View** menu item.

2. Select **Show Log Viewer**.

   The Log Viewer opens.

3. Choose the types of information to be listed (**Errors**, **Warnings**, **Informational**, and **Diagnostic**) by selecting the appropriate check boxes.

   You can select and deselect these types throughout encoding and the information will be updated automatically.

4. Choose the Functional Areas that you want the logging messages to cover.

5. After encoding, you can save the messages as a text file by clicking the **Save Messages** button.

You can also determine how many messages are stored in the log viewer, so that you only keep a reasonable limit of messages.

➤ To change Log Viewer preferences:

1. From the main window, select **Edit>Preferences**.

   The Preferences dialog opens.

2. Select the Log Viewer category.

3. Enter the number of messages you want to keep in the **Message Buffer Size** field.

4. Click **OK** to save the preference.

Using a Log File

A Log File can also be created when you run an encoding job. This file is a simple text file that contains the log messages that were created. When you create a log file, you can subsequently view and print it using a text editor.

➤ To create a Log File:

1. From the main window, select **Edit>Preferences**.

   The Preferences dialog opens.

2. Select the Log File category.

3. Choose to **Enable logging to file**.

4. Click the **Browse** button to change the directory for the log file. Or you can use the default location.

5. Enter a name for the log file, using .log as the extension.

6. You can set the maximum size for the log file by choosing to roll the file by size or by encoding time.

   New log files are created if the limit is met, preserving the old information.

7. Select the contents that you want to be logged:

   a. Choose the log format. **Short** format logs only the job name and message, and **Detailed** format also logs the category, functional area, time, and message number.

   b. Choose the types of information to be listed (**Error**, **Warning**, **Information**, and **Diagnostic**).

   c. Choose what functions to be logged from the **Functional Areas** list.

8. Click **OK** to save the preferences.

EDITING REALMEDIA FILES

RealMedia Editor allows you edit existing RealMedia files. You can edit the length of a file, cutting either the beginning or end of a clip. You can change the title, author, and other clip information. You can also can merge image maps or interactive events into a RealMedia file.

Overview: RealMedia Editor

RealMedia Editor is installed automatically when you install Helix Producer, either the Windows or Macintosh version. Click on **File>Edit RealMedia File** from the Helix Producer main window to begin.

Once you have started RealMedia Editor, the main window opens for you. Below is a brief tour of the main window, introducing each element to you.

Main Menu—the menu items at the top of the window allow you to access different functions of this program

Clip Viewer—a video window that shows the video portion of the clip

Timeline—a graphical representation of the input file. You can move the slider to any keyframe or edit point. The current time position is also shown here.

Clip Info Area—this area allows you to quickly change the title, author, copyright, keywords, and description of a clip.

Opening a RealMedia File

The first step in editing a RealMedia clip is to open it with the RealMedia Editor. Once you have opened the clip, you can play it, navigate through it, and edit it.

➤ To open a clip:

1. In the main window, choose **File>Open RealMedia File**.

   The Open RealMedia File dialog opens.

2. Navigate to the directory where the RealMedia file (a file with a **.rm** extension) is located and select it.

> **Note:** Currently, only RealMedia files with a maximum file size of 2 GB can be edited with this product.

3. Click **Open**.

The file opens in the RealMedia Editor main window. If it is a RealVideo clip, the first frame appears in the viewer.

> **Tip:** You can also drag a .rm file into the RealMedia Editor main window to open that clip instantly.

Navigating Along a Clip

RealMedia Editor gives you different methods for you to navigate along a clip so that you can find the points at which you want to edit the clip. You can use the slider on the timeline, use the Play and Stop buttons, use edit points, or use keyframes. Use one or a combination of methods to reach the point in your clip you want to edit.

Using the Timeline

The timeline is the easiest way to navigate along your clip. The red line shown in the timeline marks the current position in the clip. To move the line, either drag it with your mouse, or click with the mouse button at a point in the timeline.

When the line is at the new position, the Clip Viewer shows the current video frame.

Using Buttons

Using the navigation buttons is another way to navigate. Click the **Play** button to start the clip. And click **Stop** when you reach the desired point in the clip.

Using Keyframes

Keyframes are video frames that are encoded pixel-for-pixel into RealMedia. A RealMedia clip consists of a number of keyframes with the other in-between frames based on the information from the keyframes. Video clips must have at least one keyframe.

Click the >> button to go to the next keyframe in the clip, and click the << to go to the previous keyframe in the clip. Navigating with keyframes allows you to quickly jump to a main section of a clip.

Using Edit Points

Edit points are simply other possible points at which you can edit a clip. These points are designated by the smallest block of audio data that a clip can be split up into. There are no edit points in a video-only clip, so you can edit the clip at any point.

Click the > button to go to the next edit point in the clip, and click the < button to go to the previous edit point in the clip.

Editing with the RealMedia Editor

This section shows you how use the RealMedia Editor to edit a RealMedia clip. You will learn how to edit out the beginning or end of a clip, how to change clip information, how to merge either an image map or an interactive event to a clip, and how to append another clip to the current clip.

Editing a Clip's Beginning or End

With RealMedia Editor, you can edit out the beginning or the end of a clip. This section shows you how to mark the points for editing and gives you tips on editing out sections of your clip.

➤ To edit out the beginning or end of a clip:

1. Open a RealMedia file.

2. Navigate to the point at which you want to mark the beginning of the edited clip.

3. Click the **Mark In** button.

   On the timeline, the beginning of the clip is marked by a bracket (**[**), and the rest of the clip that remains has a darker grey color. The new beginning time of the clip is also shown.

4. Navigate to the point at which you want to mark the end of the edited clip.

5. Click the **Mark Out** button.

On the timeline, the end of the clip is marked by a bracket (**]**), and the clip that is between the two brackets has a darker grey color. The new ending time of the clip is also shown.

6. Click the **Play Selection** button to play the marked section of the clip and verify that the clip is edited properly.

7. If you are satisfied with the new clip, select **File>Save RealMedia File As** and give the newly edited file a new name.

Editing Tips

Follow these tips to get the best out of your edited clip:

• If you mark an image in between keyframes, the resulting video may have a black or frozen image, or it may simply begin at the next keyframe.

• Video clips must contain at least one keyframe. If you create a clip without a keyframe, you will be unable to save it.

• You can type in times directly for current clip position using the format: `days:hours:minutes:seconds.milliseconds`.

Changing Clip Information

Clip information tells the audience about a clip and allows the audience to find a clip more easily. With RealMedia Editor, you can add clip information if none exists or you can change clip information that is attached to a file.

➤ To change clip information:

1. Open a RealMedia file.

2. In the Clip Info area, enter new information for the clip:

• **Title**—the title of the clip

• **Author**—the name of the clip's creator

• **Copyright**—year and owner of the clip's copyright

• **Keywords**—words that you think your audience would use to search for your clip.

• **Description**—a brief summary of the clip's contents

3. To enter more information, click the **Clip Info** button.

The Edit Clip Info and File Properties dialog opens.

4. You can also select a content rating. You have the following choices:

- **No Rating**
- **All Ages**
- **Older Children**
- **Younger Teens**
- **Older Teens**
- **Adult Supervision Recommended**
- **Adults Only**

5. Click **OK** to return to the main window.

6. To save the new clip, select **File>Save RealMedia File As** and give the newly edited file a new name.

Merging Image Maps or Events

Image maps are text files that create clickable fields within a RealMedia file. When clicked by the audience, a certain action—such as opening a company's Web site—occurs. Events are similar to image maps, except the action happens automatically during playback without any audience interaction.

For an image map or an event to work during playback, they must be "merged" to a RealMedia file. RealMedia Editor gives you the ability to merge these files to a RealMedia file.

➤ To merge an image map into a RealMedia file:

1. Open a RealMedia file.

2. Select **Merge Image Maps** from the **Tools** menu.

    The Merge Image Map File dialog opens.

3. Locate the image map text file that you will merge, select it, and click **Open**.

    RealMedia Editor merges the image map to the RealMedia file, creating a new, "Untitled" clip; the original clip is closed.

➤ To merge an event into a RealMedia file:

1. Open a RealMedia file.

2. Select **Merge Events** from the **Tools** menu.

    The Merge Events File dialog opens.

3. Locate the events text file that you will merge, then select it and click **Open**.

   Helix Producer merges the event to the RealMedia file, creating a new, "Untitled" clip; the original clip is closed.

Appending Other Clips to Your Clip

Helix Producer also allows you to combine one RealMedia file to another, thus "appending" a file to the current file. Appending only works when the number and type of streams in each file is identical. Therefore, this feature works best when the source files are all encoded with the Helix Producer using the exact same settings.

Because of RealMedia Editor's 2 GB limit for the size of files it can open, the combination of appended files can not be over this limit.

➤ To append a file:

1. Open a RealMedia file.

2. Select **Append RealMedia File** from the **File** menu.

   The Open RealMedia File dialog opens.

3. Locate the file you want to append, then select it and click **Open**.

   RealMedia Editor appends the file to the currently open file. This process can take some time, depending on how large the appended file is.

4. When the appending process is complete, you can edit the combined files as a new clip; the original clip is closed.

Advanced Editing Features

In addition to editing files, RealMedia Editor gives you advanced features that can help you with the editing process. You can view information about all streams that are incorporated into a RealMedia file. Plus, you can decide how fast RealMedia Editor will preview a clip: quickly or normal speed.

Viewing Stream Information

Every RealMedia file is created with a number of streams. Single rate files contain only one stream, while SureStream files can contain multiple streams depending on the number of target audiences a clip is encoded for.

RealMedia Editor allows you to easily view valuable information about each stream that associated with a RealMedia file.

➤ **To view stream information:**

1. Open a RealMedia file.

2. Select **Stream Info** from the **Tools** menu.

   The Stream Info dialog opens, giving you the following information about the file:

   • **Audio Mode**—the type of audio for the encoded clip (such as music or voice)

   • **Video Mode**—the quality of the encoded video (such as normal or smooth motion)

   • **Video Size**—the size and width, in pixels, of the encoded video

   • **File Type**—either single rate or SureStream

   This dialog also gives you the following information about each stream within the file:

   • **Bit Rate**—the amount of bandwidth necessary to view the stream

   • **Preroll**—the amount of bits needed to download before the stream can play

   • **Frequency**—the frequency response of the encoded audio for the stream

   • **Codec**—the compression/decompression coded used to encode the stream.

3. Click **Close** to return to the main window.

Changing RealMedia Editor Preferences

RealMedia Editor allows you to change how you preview a clip when it is opened in the main window. You can set RealMedia Editor to view a clip quickly, or you can set RealMedia Editor to view a clip more accurately. If you choose to view a clip accurately, it may take much more time to open if the clip is large in size.

➤ **To change Helix Producer preferences:**

1. Select **Preferences** from the **Tools** menu.

   The Preferences dialog opens.

2. Select either **Accurate** or **Quick** to designate how you want Helix Producer to open new RealMedia files.

 • **Quick**—plays back the selection immediately with out delay, but audio and video may not play exactly as they would in the saved clip. For example, frames of video at the beginning of a clip may appear different. And audio may not stop precisely at the end of a clip.

 • **Accurate**—saves the selection to a temporary file and can take longer for long selections. The preview, however, will look and sound exactly like the saved clip.

3. Click **OK** to return to the main window.

USING THE COMMAND LINE

This chapter shows you how to run Helix Producer from the command line. The Helix Producer command line application gives you the ability to encode streaming media, both as files and as a live broadcast. You will also learn about other installed utilities that allow you to use the command line to create and modify streaming media files.

Overview: Command Line Programs

The following command line programs are available to you:

- The **producer.exe** command line application allows you to create RealMedia files and broadcast to a server from RealNetworks. Using commands, you have the ability to set up input devices, use video prefilters, select destinations, and change audience settings. For more information, see "Using the producer.exe Application" below.

- The **rmeditor.exe** utility allows you to change information about a RealMedia file, trim the length of the clip, combine multiple RealMedia files, and create a text file containing the file's information. For more information, see "Using the rmeditor.exe Utility" on page 107.

- The **rmevents.exe** utility allows you to merge image map and event text files into an existing RealMedia file, and extract events and image maps from a RealMedia file into a text file. For more information, see "Using the rmevents.exe Utility" on page 108.

Using the producer.exe Application

The **producer.exe** application is fully functional, giving you the same encoding capabilities as the Helix Producer main interface. This application is also designed for users who want to encode a batch of files using wildcards.

There are two ways to use this application. The first way to use **producer** is to run the program, typing in options and their values for the different aspects of the encoding process. The second way is to create a job file and run the program with that job file. A job file is a saved text file that shows Helix Producer how to run an encoding session.

The rest of this section shows you the full syntax of the **producer.exe** application, then breaks down all of the options into different sections, giving a description of each option. Finally, some examples of common uses are provided.

producer.exe Syntax

This section shows you the basic syntax to encode streaming media with the **producer** command-line application. It also gives you a brief description of the options you can use with the application. Use the references to find more detailed information about these options, located later in this chapter.

The basic syntax of the **producer.exe** command-line application is as follows:

producer [*input*] [*output*] [*general_options*]

Where [*input*] is one of the following:

producer.exe Command Line Input Options

Option	Value	Function	Reference
-ac	*audioDeviceID*	Specifies the audio capture input device ID.	page 93
-ap	*audioPort*	Indicates the port of the audio capture input.	page 93
-i	*filename.ext*	Sets the input file name.	page 93
-vc	*videoDeviceID*	Specifies the video capture input device ID.	page 93
-vp	*videoPort*	Indicates the port of the video capture input.	page 93
-vf	*videoFormat*	(Linux Only.) Specifies the video format of the video capture device.	page 94

And where [*output*] is one or more of the following:.

producer.exe Command Line Output Options

Option	Value	Function	Reference
-o	*filename*.rm	Sets the output file name.	page 94
-sg	Username:Password@Server:Port/ streamname	Sets server definition using account-based encoder protocol.	page 95
-si	Password@Server:Port/ streamname	Gives settings required for a server-initiated broadcast output stream using the Pull Broadcast method.	page 95
-sp	Password@Server:PortRange/ streamname	Gives settings required for live server output using the Push Broadcast method.	page 94
-sd	[username:password@]ServerDefi nition,streamname	Uses a pre-existing Server Destination as the output.	page 96

And where [*general_options*] are any of the following:

producer.exe Command Line General Options

Option	Value	Default	Function	Reference
-a	*text*	(none)	Encodes an author name.	page 100
-ad	*audience_names*	16k, 28k, 56k, and 256k	Specifies audience encoding speeds.	page 96
-ag	-12 to 12	0	Uses the audio gain filter.	page 99
-am	voice|music	voice	Sets the audio type.	page 97
-arq	fast|high	high	Sets the type of audio resampling used.	page 98
-bl	(none)	true	Applies the black level filter.	page 99
-c	*text*	(none)	Sets the encoded copyright.	page 100
-cj	*filename*.rpjf	(none)	Creates a job file.	page 104
-cr	*left,top,width, height*	(none)	Sets cropping parameters in pixels.	page 99
-d	dd:hh:mm:ss.xyz	*infinite*	Sets total time for encoding.	page 94
-de	*text*	(none)	Sets clip description.	page 100

(Table Page 1 of 2)

producer.exe Command Line General Options (continued)

Option	Value	Default	Function	Reference
-di	auto|d|i|both	(none)	Applies deinterlace or telecine filters.	page 99
-dlf	(none)	(none)	Disables logging to file.	page 101
-dls	(none)	(none)	Disables logging to screen.	page 101
-k	*text*	(none)	Adds keywords.	page 100
-lc	error|warning| info|diagnostic	(none)	Specifies a logging category.	page 101
-nf	low|high	(none)	Applies the video noise filter.	page 99
-q	(none)	(none)	Disables all screen output.	page 101
-rq	fast|high	fast	Specifies the resize quality.	page 98
-rs	*widthxheight*	0x0	Sets the resizing dimensions.	page 98
-t	*text*	(none)	Adds a title.	page 100
-dt	(none)	(none)	Disables two-pass encoding.	page 97
-da	(none)	(none)	Disables audio from the input.	
-dv	(none)	(none)	Disables video from the input.	
-vm	sharp|normal| smooth|slideshow	normal	Sets the video mode.	page 97

(Table Page 2 of 2)

You can also use the application to display information without encoding streaming media:

producer [*information_options*]

Where [*information_options*] is one of the following:.

producer.exe Command Line Information Options

Option	Function	Reference
-pa	Prints audience definitions for use with the -ad option.	page 102
-pd	Displays device information for use with the -d option.	page 102
-ps	Prints server definitions for use with the -sd option.	page 102
-v	Displays the Helix Producer version number.	page 102

Input and Output Options

The following sections describe in more detail the input and output options used with the **producer** command-line application.

Input File (-i)

The input file to be encoded, located on a network or local drive. Wildcards can be used to run a batch of jobs. The value is a valid file name of a job file. You can use either a relative path from the working directory, or an absolute path. Here are some examples:

```
-i myfile.avi
-i c:\files\movie.avi
-i c:\files\*.avi
```

Audio Capture Device ID (-ac)

Specifies the audio device ID for the input audio devices. Using this option overrides the -i option. Use the Print Device Information (-pd) option to list values on your system. You can also use wildcards. Here are some examples:

```
-ac 0
-ac "Sound Blaster*"
```

Audio Capture Device Port (-ap)

Specifies the audio port for the input audio devices. Overrides the -i option. Use the Print Device Information (-pd) option to list values on your system. For example:

```
-ap "mic"
```

Video Capture Device ID (-vc)

Specifies the device ID for the input video device. Overrides the -i option. Use the Print Device Information (-pd) option to list values on your system. Here are some examples:

```
-vc 0
-vc "Osprey*"
-vc /dev/video3
```

Video Device Port (-vp)

Specifies the device port for the input video capture device. If omitted, the current mixer settings are used. Overrides the -i option. Use the Print Device Information (-pd) option to list values on your system. You can also use wildcards. Examples:

```
-vp 0
-vp "Composite 1"
```

Video Format (-vf)

Linux only option. Specifies the video format of the input video capture device. Value can be NTSC, PAL, SECAM, AUTO, or an integer. Examples:

```
-vf SECAM
-vf 8
```

Audio/Video Device Duration (-d)

Specifies the amount of time before the job ends and encoding from the device stops. Duration can be specified in `Days:Hours:Minutes:Seconds`:

```
[d:][h:][m:][s[.ms]]
```

The default is an infinite duration. Here are some examples:

```
-d 90 (90 seconds)
-d 1:00:00 (1 hour)
```

Output File or Directory (-o)

The file name for the output RealMedia file, or the directory if multiple input files are used (batch encoding). If multiple files are used, the output names are automatically generated. If the specified output directory does not exist, it will be created.

The general file extension for RealMedia files is **.rm**. If you are encoding with Variable Bit Rate (VBR), you must use the **.rmvb** extension.

Examples:

```
-o C:\files\movie.rm
-o C:\output\variable.rmvb
-o C:\Movies
```

Push Server (Server 9) (-sp)

Settings required for a live server output using the Push Broadcast method available in Helix Universal Server 9 and greater. Value is:

```
[password@]address[:startPort[-endPort]]/streamname
```

Where:

password	Password for encoding to server. Required. If included, the authentication type is assumed "Basic".
address	Any valid IP address or domain name.

startPort	Start port on the remote server to broadcast packets to. Must be an integer from 1 to 65535. Default is 30000.
endPort	End port on the remote server to broadcast packets to. Default value is same as startPort.
streamname	Any valid stream name.

Example:

`-sp pass@server.co.com:3000-30001/live.rm`

Pull Server (Server 9) (-si)

Settings required for a server-initiated broadcast output stream using the Pull Broadcast method available in Helix Universal Server 9 and greater. Value is:

`[password@]listenAddress:listenPort/streamname`

Where:

password	Password for encoding to server.
listenAddress	Any valid IP address. This is the address which the producer will bind to for receiving pull server requests from remote servers.
listenPort	Listen port on the encoder to listen on. Must be an integer from 1 to 65535. Default is 3031.
streamname	Any valid stream name.

Examples:

```
-si pass@192.168.224.221:3031/live.rm
-si 205.198.34.3/wildlife.rm
```

Legacy Push Server (-sg)

Settings required for encoding to RealServer versions G2 through 8.01. Value is:

`[username:password@] address[:port]/streamname`

Where:

username	Username for encoding to server.
password	Password for encoding to server.
address	Any valid IP address or domain name.
port	Port on he remote server to broadcast packets to. Must be an integer from 1 to 65535. Default is 4040.
streamname	Any valid stream name.

Examples:

`-sg name:pass@realserver.yourcompany.com/live.rm`

`-sg jdoe:letmein@172.16.0.34:4040/meeting.rm`

Server Definition or Server File (-sd)

May be either the name of a Server defined within the Helix Producer or a Server File located anywhere on the hard drive. If a Server File is referenced, the full filename must be included, including file extension. The **producer** application will search for a filename matching the filename first and then look in the Servers defined within the Helix Producer. If relative paths provided for server files, the location is relative to the working directory. Value is:

`[username:password@]ServerDefinition,streamname`

Where:

ServerDefinition	A string representing a Server Definition defined within the Helix Producer or the name of a Server File with extension.

Examples:

```
-sd janedoe:letmein@RBN Hosting Server,live.rm
-sd Server1,/live/radio.rm
-sd janedoe:letmein@C:\settings\Server1.rpsd/live.rm
```

General Options—Encoding Options

You can use the following encoding setting options with the **producer** utility.

Audience Definitions or Audience Files (-ad)

This option provides a list of Audience Definitions defined within the Helix Producer. The name of the audience may be abbreviated with the first few letters. The closest matching audience is used. For example, "28k Dial-up" may be abbreviated with "28." The default is 28k Dial-up and 56k Dial-up. The value is a comma-separated list of strings that matches the names of an Audience Definitions defined within the Helix Producer.

You can also use an Audience File that you have created. An Audience File is an XML-based file that specifies the video codec, audio codec, bit rate and other settings for the audience. If you use an Audience File, you must enter the full

filename, including the extension. For information on using Audience Files, see "Creating an Audience File" on page 104.

Here are examples:

```
-ad "28k,56k"
-ad 150k
-ad "28k Dial-up,56k Dial-up,256k DSL/Cable Modem"
-ad "C:\settings\My 28k Modem.rpad"
-ad "/files/28k.rpad,/files/56k.rpad"
```

Audio Mode (-am)

Describes the type of audio content. This setting determines which audio codec is to be assigned from an audience. Values are voice or music. The default is voice. Example:

```
-am music
```

Video Mode (-vm)

Setting to control the picture quality of the encoded video. Values include sharp, normal, smooth, and slideshow. The default is normal. For example:

```
-vm sharp
```

Disable Two-Pass Encoding (-dt)

The input file is read in two passes, the first to analyze, and the second to encode. Only applies for encoding file-to-file. It is ignored if the input is a device (-d) or the output is a server (-sg, -sp, -si, -sd). No value is required. For example:

```
-dt
```

Disable Audio (-da)

If used, the audio from the input source will be stripped out. No value is required. For example:

```
-da
```

Disable Video (-dv)

If used, the video from the input source will be stripped out. No value is required. For example:

```
-dv
```

Resize Quality (-rq)

Affects the resulting quality of an output file when resize is applied. Choosing high results in a better quality resize, but uses considerably more CPU. Ignored if the -rs option is omitted. Values are fast (default) and high. Example:

```
-rq high
```

Resize Video (-rs)

Sets the output video width and height in pixels. If either is set to 0, its value is computed from the original video's aspect ratio. Value is:

```
WxH
```

Where:

W	Width of the video in pixels from 0-2048. Must be divisible by 4.
H	Height of the video in pixels from 0-2048. Must be divisible by 4.

Examples:

```
-rs 320x240
-rs 320x0 (resize width to 320 and maintain aspect ratio)
-rs 0x240 (resize to 240 high but maintain aspect ratio)
-rs 0x240 (resize to 240 high but maintain aspect ratio)
```

Audio Resampling Quality (-arq)

Allows you to specify the quality of audio resampling. Resampling must be done if the sample rate required by the target audio codec is different from the input file or device sample rate. The producer automatically invokes the resampler for each audio codec as needed.

Values are high or fast. Choosing high results in an audio-only encode using as much as 2 times as much CPU for live broadcasting or taking twice as long to encode for file-to-file encoding. Choosing fast does not incur additional CPU but the encoded output will have a slightly lower quality. In all cases, there is no pitch shift incurred by the audio resampler.

For example:

```
-arq fast
```

General Options—Prefilter Options

The following prefilter options are available for the **producer** utility.

Audio Gain Filter (-ag)

Prevents the audio signal from clipping by applying dynamic range compression, as well as offering gain control to increase or decrease the gain of the signal. The value is any number from -12.0 to 12.0. For example:

```
-ag 0
```

Black Level Filter (-bl)

This filter is designed to make black colors darker. If the source video has a washed-out appearance, use this filter to help restore proper color. No value is required. Example:

```
-bl
```

Inverse Telecine and Deinterlace Filters (-di)

Specifies use of inverse telecine and/or deinterlace filters. If you are unsure whether input was created using the telecine or interlace processes, you can specify autodetect. Values is `auto`, `d`, `i` or `both`, where:

`auto`	Autodetect and apply De-interlace and Inverse Telecine filters as needed.
`d`	Use De-interlace filter only.
`i`	Use Inverse Telecine filter only.
`both`	Apply both de-interlace and Inverse Telecine.

Example:

```
-di auto
-di i
```

Crop Video Input (-cr)

Prefilter that crops out unwanted video. Values are pixel values for left, top, width, and height. For example:

```
-cr 2,2,320,240
```

Video Noise Filter (-nf)

Removes small distortions or static in the video image. Value is `low` or `high`. Examples:

```
-nf low
-nf high
```

General Options—Clip Information Options

The following clip information options are available for use with the **producer** utility.

Title (-t)

The title for the recorded clip. Use quotes if spaces are used. The value can be any string up to 255 characters. Example:

```
-t "A Slug's Life"
```

Author (-a)

The author of the recorded clip. Use quotes if spaces are used. The value can be any string up to 255 characters. Example:

```
-a "Billy Shakespeare"
```

Copyright (-c)

The copyright owner and date for the recorded clip. Use quotes if spaces are used. The value can be any string up to 255 characters. Example:

```
-c "Diznee 2001"
```

Keywords (-k)

Keywords that will help search engines locate your clip. Use quotes if spaces are used. The value can be any string up to 1023 bytes. Example:

```
-k "pets slugs"
```

Description (-de)

A brief description of the clip. Use quotes if spaces are used. The value can be any string up to 64 kb. Example:

```
-de "a gripping story about a man and his pet banana slug
```

Content Rating (-r)

Defines the age range for which the content is applicable. Available options are: 0 - No Rating , 1- All Ages 2 - Older Children, 3 - Younger Teens, 4 - Older Teens (15 & Up), 5 - Adult Supervision Recommended, 6 - Adults Only. Example:

```
-r 1
```

General Options—Screen Output Options

You can use the following screen output options with the **producer** utility.

Logging Category (-lc)

Determines the level of logging messages used. Informational messages are low volume and high importance while diagnostic messages are high volume and generally low importance. Values are error, warning, info, and diagnostic. Value may be abbreviated. Examples:

```
-lc "error,warning,diagnostic"
-lc "err,warn,info"
-lc "e,w,i,d"
```

Disable Logging to File (-dlf)

Suppresses logging to file. No value is required. Example:

```
-dlf
```

Disable Logging to Screen (-dls)

Suppresses logging to the screen. No value is required. Example:

```
-dls
```

Quiet (-q)

All output is disabled. Useful if you are running the producer from within another application that fails if data output to standard output or standard error. No value is required. Example:

```
-q
```

Information Options

The following information options are available for use with the **producer** utility.

Help (-h)

Returns help for using the command line producer. No value is required. Example:

```
-h
```

Print Device Information (-pd)

Extracts audio and video device information from the system for use in the `-d` option. See "Audio/Video Device Duration (-d)" on page 94 for information on how to use the information printed with this command. This option may only be used by itself on the command line. No value is required. Example:

```
-pd
```

Print Audiences (-pa)

Prints a list of accepted Audience Definitions for use in the `-ad` option. No value is required. Example:

```
-pa
```

Print Servers (-ps)

Prints a list of accepted Server Definitions for use in the `-sd` option. No value is required. Example:

```
-ps
```

Print Version Information (-v)

Prints the version of Helix Producer used. No value is required. Example:

```
-v
```

Examples

This section shows you some examples of how to put the preceding options together for some common uses of the **producer.exe** command line program.

Simple Encode

The following example is a simple encode of a media file. Using the default values for all options, the output is named file.rm and is encoded using the 28k and 56k audience definitions.

```
producer -i file.avi
```

Batch Encoding with a Wildcard

The following example encodes a batch of files using a wildcard to designate the input files. The output files are all placed in the movies directory.

```
producer -i C:\capture\*.avi -o C:\movies\
```

Live Input Capture to File

The following example captures input from the video capture card and the audio sound card and writes to output file capturefile.rm.

```
producer -vc 0 -ac 0 -o c:\capturefile.rm
```

Live Input Capture for Broadcast

The following example captures audio and video and streams to a standard Helix Universal Server using streamname live.rm.

```
producer -v 0:2 -ac 0 -s janedoe:pass@realserver.mycompany.com:4040/live.rm
```

Capture with Audience Targets

The following example captures audio and video and streams both to a server and to a file. It also targets the 56k and 150k audiences.

```
producer -v 0:2 -ac 0 -s me:letmein@realserver.abc.com:4040/live.rm
-o archive.rm -ad "56k,150k"
```

Clip Information

The following example shows you how to encode using clip information: a title, author, and copyright.

```
producer -i myfile.mov -o output.rm -t "Summer Vaction" -a "John Doe"
-k "Cape Cod Summer Vacation 2001"
-de "Video from summer vacation in Cape Cod."
```

Prefilters

The final example shows you how to use prefilters. Here, the example uses the audio gain with a value of 4, audio mode of music, resizes to 240 high by maintaining aspect ratio, uses high quality resize, applies a black level filter and performs auto-detection of De-interlace and Inverse Telecine.

```
producer -i vacation.mov -o vacation.rm -ag 4 -am music -rs 240x0 -rq high -bl -di
auto
```

Encoding With a Job File

When you use a job file, all encoding options are listed within the file, and you need to pass a maximum of two options on the command line. Use the following syntax when encoding with a job file:

```
producer -j job_filename -sa Username:Password@ServerDefinition
```

Job Filename (-f)

Tells the utility to run a Job File from the command line. The value is a valid file name of a job file. You can use either a relative path from the working directory, or an absolute path. For example:

```
-j MyJob.rpjf
```

Create Job File (-cj)

Generates a Job File from the command line options provided. The value is a valid file name of a job file. You can use either a relative path from the working directory, or an absolute path. For example:

```
-cj NewJob.rpjf
```

Creating an Audience File

Audience Files are XML-based files that you edit using any text editor. These files can be used with the command line application to customize the audiences that you will use for encoding.

Helix Producer is installed with a number of default Audience Files that you can use to modify and create your own. These default files are located in the Audiences directory off of the application directory. It is recommended that if you modify a default audience file, you save it under a different filename.

Changing RealAudio Stream Sections in an Audience File

Within the Audience File, there is a section for each audio stream used. For each stream you specify the codec name, codec flavor, and the context of the codec used.

Below are tables that lists codec names and flavors, and also lists the corresponding RealAudio codec. See "Appendix A: RealAudio Codec Reference" beginning on page 113 for more information on RealAudio codecs.

Table 1: Music RealAudio Codecs

Name	Flavor	RealAudio Codec Used
cook	8	6 kbps Music
cook	0	8 kbps Music
cook	1	11 kbps Music
cook	26	12 kbps Stereo Music, RA 8
cook	2	16 kbps Music

(Table Page 1 of 3)

Table 1: Music RealAudio Codecs (continued)

Name	Flavor	RealAudio Codec Used
cook	17	16 kbps Stereo Music, RA8
cook	3	20 kbps Music
cook	15	20 kbps Music—High Response
cook	9	20 kbps Stereo Music
cook	18	20 kbps Stereo Music, RA8
cook	19	20 kbps Stereo Music—High Response, RA8
cook	4	32 kbps Music
cook	16	32 kbps Music—High Response
cook	10	32 kbps Stereo Music
cook	20	32 kbps Stereo Music, RA8
cook	21	32 kbps Stereo Music—High Response, RA8
cook	5	44 kbps Music
cook	11	44 kbps Stereo Music
cook	22	44 kbps Stereo Music, RA8
cook	23	44 kbps Stereo Music—High Response, RA8
cook	29	44 kbps Surround Audio
cook	6	64 kbps Music
cook	12	64 kbps Stereo Music
cook	24	64 kbps Stereo Music, RA8
cook	27	64 kbps Surround Audio
atrc	0	66 kbps Stereo Music, RA8
atrc	1	94 kbps Stereo Music, RA8
cook	13	96 kbps Stereo Music
cook	25	96 kbps Stereo Music, RA8
cook	28	96 kbps Surround Audio
atrc	2	105 kbps Stereo Music, RA8
atrc	3	132 kbps Stereo Music, RA8
atrc	8	132 kbps Surround Audio
atrc	4	146 kbps Stereo Music, RA8

(Table Page 2 of 3)

Table 1: Music RealAudio Codecs (continued)

Name	Flavor	RealAudio Codec Used
atrc	9	146 kbps Surround Audio
atrc	5	176 kbps Stereo Music, RA8
atrc	10	176 kbps Surround Audio
atrc	6	264 kbps Stereo Music, RA8
atrc	11	264 kbps Surround Audio
atrc	7	352 kbps Stereo Music, RA8
atrc	12	352 kbps Surround Audio

(Table Page 3 of 3)

Table 2: Voice RealAudio Codecs

Name	Flavor	RealAudio Codec Used
sipr	2	5 kbps Voice
sipr	0	6.5 kbps Voice
sipr	1	8.5 kbps Voice
sipr	3	16 kbps Voice
cook	7	32 kbps Voice
cook	14	64 kbps Voice

Changing the RealVideo Stream Section in an Audience File

Within the Audience File, there is a section for the video stream. Here, you specify the following:

- **codecName**—the RealVideo codec name used to encode the video. Below is a table that lists codec names and the corresponding RealVideo codec.

Table 3: RealVideo Codecs

Name	RealAudio Codec Used
rv9	RealVideo 9
rv8	RealVideo 8
rvg2svt	RealVideo G2 with SVT

- **encodingType**—the type of encoding used can be Constant Bit Rate (**cbr**), Variable Bit Rate using quality of video as a goal (**vbrQuality**), or Variable Bit Rate using an average bit rate as a goal (**vbrBitrate**). If you use Variable

Bit Rate, you can only use one audience and the output file name must use a **.rmvb** extension.

- **quality**—the target quality percentage that the video codec will attempt to encode the video at. Used if you specify the **encodingType** to be **vbrQuality**.

- **maxStartupLatency**—the maximum time (from **4** to **25** seconds) that the clip will buffer before playing begins.

- **maxFrameRate**—the ideal, maximum frame rate (up to **30** frames per second) that the video codec will attempt to encode the output video at.

- **maxKeyInterval**—the time (from **0** to **60** seconds) in between key frames of encoded video.

- **enableLossProtection**—whether or not (**true** or **false**) to add error correction information to the encoded file.

Using the rmeditor.exe Utility

The **rmeditor.exe** utility allows you to edit a RealMedia file, or get information about an encoded file. There are three ways to use this utility. The first is to get information about a RealMedia file (also called a "dump"). The second is to change metadata, such as title, author, and copyright information. The third way to use this utility is to cut files and paste two files together. The following table summarizes the **rmeditor** options, which have both long and short option flags.

rmeditor.exe Utility Options

Flag (long and short)		Value	Function
-input	-i	*filename.ext*	Indicates path to the input file.
-output	-o	*filename*.rm	Specifies path to the output file.
-title	-t	*text*	Gives the title text.
-author	-a	*text*	Specifies the author name.
-copyright	-c	*text*	Provides the copyright text.
-comment	-C	*text*	Any comments about the clip.
-startTime	-s	*dd:hh:mm:ss.xyz*	Sets clip start time.
-endTime	-e	*dd:hh:mm:ss.xyz*	Sets clip end time.
-logFile	-l	*filename*.log	Gives path to the log file.
-dumpFile	-d	*filename*.txt	Sets the path to the dump file.

(Table Page 1 of 2)

Flag (long and short)		Value	Function
-description	-q	*text*	Describes the file contents.
-keywords	-n	*text*	Provides a list of keywords.

(Table Page 2 of 2)

Getting Information from a RealMedia File

Use the following syntax to create a text-based dump file that contains information about an encoded RealMedia file:

rmeditor -inputFile *filename*.rm -dumpFile *filename*.txt

Editing Metadata

rmeditor {-inputFile <filename>} {-outputFile <filename>}

    [-t | -title <title>] [-a | -author <author>] [-c | -copyright <copyright>]

    [-q | -description <description>] [-n | -keyword <keywords>]

Cutting and Pasting Files

rmeditor {-i | -input | -inputFile <input> [-s | -startTime <dd:hh:mm:ss.xyz>] [-e | -endTime <dd:hh:mm:ss.xyz>] }

    [-i | -input | -inputFile <input> [-s | -startTime <dd:hh:mm:ss.xyz>] [-e | -endTime <dd:hh:mm:ss.xyz>]]

    ...

    {-o | -output | -outputFile <filename>}

    [-t | -title <title>] [-a | -author <author>] [-c | -copyright <copyright>]

    [-q | -description <description>] [-n | -keyword <keywords>]

Using the rmevents.exe Utility

The **rmevents.exe** utility lets you encode HTML-page URLs into a RealMedia clip. You can even target a specific HTML pane in RealOne Player's three-pane environment. Using **rmevents.exe**, you can also encode clip information such as title, author, and copyright, and create image maps with clickable hot spots in different shapes.

> **For More Information:** To learn more about RealOne Player's three-pane environment, download *Introduction to Streaming*

Media with RealOne in PDF or zipped HTML format from
http://service.real.com/help/library/encoders.html.

Writing an Events File

You first create an events file, which is a plain text file that uses the standard
text extension of .txt. This file describes events that occur as the clip plays. You
write each event on a separate line, and you can use a pound sign (#) to start a
comment line. Each event line follows this format:

flag start_time end_time event_syntax

The flag indicates the type of event, which can be either to display clip
information, or to open a URL in RealOne Player HTML pane automatically.
The starting time and ending time are relative to the start of clip playback.
You indicate the time value with the following format, in which only the
seconds field is required:

dd:hh:mm:ss.x

> **Note:** Define each event on a single line within the events text
> file. Do not press **Enter** to wrap long lines manually.

Specifying URL Events

When opening a URL automatically in a RealOne Player HTML pane, you use
the u event flag. The event syntax looks like this:

u *start_time end_time* &&*target*&&*URL*?*parameters*

The URL must be a fully qualified HTTP URL. For the parameters, you can
use rpcontextheight="*pixels*" and rpcontextwidtht="*pixels*" when sending a URL
to the related info pane. For *target*, use one of the following:

_rpcontextwin	Display the URL in the related info pane.
_rpbrowser	Display the URL in the media browser pane.
_rpexternal	Display the URL in a secondary browsing window.

The following is a sample events file that opens two URLs in the related info
pane at different times, and two URLs in the media browser pane at different
times:

```
# Open a URL in the related info pane when the clip starts, and size the pane.
u 00:00:00.0 00:01:59.9 &&_rpcontextwin&&
http://www.example.com/info1.html?rpcontextheight=250&rpcontextwidth=280
#
# Open a URL in the media browser pane at the 1-minute mark.
u 00:01:00.0 00:01:59.9 &&_rpbrowser&&http://www.example.com/index.html
#
# Open a second URL in the related info pane at the 2-minute mark.
u 00:02:00.0 00:04:00.0 &&_rpcontextwin&&http://www.example.com/info2.html
#
# Open a second URL in the media browser pane at the 3-minute mark.
u 00:03:00.0 00:04:00.0 &&_rpbrowser&&http://www.example.com/index2.html
```

Note the following about this sample:

- When you open multiple URLs, list the events in ascending order according to the start times.

- The related info pane size is set for the duration of the clip by the first URL that opens in that pane. Any subsequent URLs that target the related info pane therefore do not require sizing information.

- The end times indicate a point past which the URL should not open. In the second event defined above, the URL is scheduled to open at 1:00 minutes, but no later than 1:59.9 minutes. If a viewer starts the clip and immediately seeks to its 3-minute mark, for example, the URL doesn't display because the clip never plays at any point between the URL's start and end times.

Adding Clip Information

Using **rmevents.exe**, you can encode the RealOne Player clipinfo parameters into the clip. Separate the subvalues with vertical lines, or "pipes." For example, the following shows an events file line that specifies clip information, using i as the event flag:

```
i 00:00:00.0 00:00:10.0 clipinfo:title=My Presentation|artist name=Pat
Morales|...(Does not work with Player version 8)
```

For clip information, you can specify the start time as the clip's starting time (00:00:00.0). Specifying an end time is required, but the actual end time doesn't matter because the clip information will display throughout the length of the clip playback. In the example above, the end time is set to 10 seconds after the clip starts.

The following table describes the name and value pairs that you can use with clipinfo. You can use any set of values, and list them in any order. Most text values can be over 100 characters long.

Clipinfo Parameter Values

Name and Value	Function
title=*text*	Gives the clip title.
artist name=*text*	Indicates the artist name.
album name=*text*	Gives the album name. If you specify an album name and do not also display an HTML page in the related info pane, RealOne Player displays in that pane a standard page that lists the artist, album, year, and genre values. The viewer can hide this information, though, with **Tools>Album Info>Hide**.
genre=*text*	Indicates the clip genre, such as Rock or Jazz.
copyright=*text*	Gives the copyright notice.
year=*text*	Indicates the year the content was released.
cdnum=*number*	Supplies the CD track number.
comments=*text*	Provides any additional comments.

Note: Do not use the title, author, and copyright parameters along with clipinfo.

Merging the Events File with the Clip

After you write your events file, you use the **RMEvents** utility to merge the events file with your clip.

➤ **To merge an events file with a RealAudio or RealVideo clip on Windows:**

1. Open a command line prompt from the **Start** menu.

2. Move to the main Helix Producer directory. The default main directory is c:\Program Files\Real\Helix Producer Basic or c:\Program Files\Real\Helix Producer Plus.

   Here is an example:

   C:\>cd "C:\Program Files\Real\Helix Producer Basic"

3. Enter the following command, which uses three flags to indicate the input clip, output clip, and events file:

   rmevents -i *input*.rm -o *output*.rm -e *events*.txt

where:

- *input*.rm is the path and name of the input clip

- *output*.rm is the path and name of the output clip

- *events*.txt is the path and name of the events file

> **Tip:** Always choose a new output name so that you can save your original clip without any encoded events.

REALAUDIO CODEC REFERENCE

This appendix provides a reference for all RealAudio codecs used by Helix Producer, broken down into separate tables for voice, mono music, and stereo music codecs.

Using the RealAudio Codec Reference Tables

The following are descriptions of the different columns within the codec reference tables. Each column

RealAudio Codec

This column shows the name of the codec as it appears in the Audience Templates dialog. The name gives you vital information about the codec: the bit rate for the codec, the type of audio the codec is suited for, and if the codec is a "high response" codec.

High response codecs cover high frequency sounds, such as a flute, than normal codecs at the same bit rate. But they don't cover low frequencies, such as a bass drum, as well as normal codecs.

Sampling Rate

The Sampling Rate column is the codec's optimum sample rate. Using a codec's optimum sampling rate in your audio source file ensures that the audio stays synchronized with other media and prevents pitch shifting during audio resampling. Audio quality degrades if you use lower than the optimum sampling rate.

If you use a higher sampling rate when creating the source audio, it is best to use a multiple of the optimum rate. If the optimum rate is 8 kHz, for example, use a higher rate of 16 kHz or 32 kHz. When in doubt, use a CD-quality sampling rate of 44.1 kHz for your source audio.

> **For More Information:** The audio preparation chapter of *RealNetworks Production Guide* has an expanded list that covers

all RealAudio codecs, including obsolete codecs no longer used by Helix Producer.

Voice Codecs

Helix Producer uses a voice codec when you encode a voice-only or voice-with-music clip. The lowest-speed voice codec normally used with RealAudio is 16 kbps. The lower-speed codecs can be used as "duress" streams in SureStream clips. They are also used to encode soundtracks for low-bandwidth RealVideo clips.

RealAudio Voice Codecs

RealAudio Codec	Sampling Rate
5 kbps Voice	8 kHz
6.5 kbps Voice	8 kHz
8.5 kbps Voice	8 kHz
16 kbps Voice	16 kHz
32 kbps Voice	22.05 kHz
64 kbps Voice	44.1 kHz

Stereo Music Codecs

Helix Producer contains an array of RealAudio codecs that are optimized for stereo music. These RealAudio codecs allow you to stream high-quality stereo music at a wide range of bit rates.

RealAudio Stereo Music Codecs

RealAudio Codec	Sampling Rate
12 kbps Stereo Music, RealAudio 8	11.025 kHz
16 kbps Stereo Music, RA8	22.05 kHz
20 kbps Stereo Music	11.025 kHz
20 kbps Stereo Music, RA8	22.05 kHz
20 kbps Stereo Music—High Response, RA8	22.05 kHz
32 kbps Stereo Music	22.05 kHz
32 kbps Stereo Music, RA8	22.05 kHz
32 kbps Stereo Music—High Response, RA8	44.1 kHz

(Table Page 1 of 2)

RealAudio Stereo Music Codecs (continued)

RealAudio Codec	Sampling Rate
44 kbps Stereo Music	22.05 kHz
44 kbps Stereo Music, RA8	44.1 kHz
44 kbps Stereo Music—High Response, RA8	44.1 kHz
44 kbps Surround Audio	44.1 kHz
64 kbps Stereo Music	44.1 kHz
64 kbps Stereo Music, RA8	44.1 kHz
64 kbps Surround Audio	44.1 kHz
66 kbps Stereo Music, RA8	44.1 kHz
94 kbps Stereo Music, RA8	44.1 kHz
96 kbps Stereo Music	44.1 kHz
96 kbps Stereo Music, RA8	44.1 kHz
96 kbps Surround Audio	44.1 kHz
105 kbps Stereo Music, RA8	44.1 kHz
132 kbps Stereo Music, RA8	44.1 kHz
132 kbps Surround Audio	44.1 kHz
146 kbps Stereo Music, RA8	44.1 kHz
146 kbps Surround Audio	44.1 kHz
176 kbps Stereo Music, RA8	44.1 kHz
176 kbps Surround Audio	44.1 kHz
264 kbps Stereo Music, RA8	44.1 kHz
264 kbps Surround Audio	44.1 kHz
352 kbps Stereo Music, RA8	44.1 kHz
352 kbps Surround Audio	44.1 kHz

(Table Page 2 of 2)

Stereo Music Basics

The biggest difference between stereo and mono music is that mono music has one channel that gets sent to both speakers. Stereo music has separate channels for the left and the right. That means a stereo version of an audio clip needs to hold more data than a mono version. Yet Helix Producer can make stereo and mono clips for the same bandwidth. How can it do this?

The answer lies in the codecs' frequency responses. Remember, RealAudio is a lossy compression scheme that throws out data. The stereo codec squeezes

both channels down to the same size as the mono codec by throwing out more data for each channel. This makes its frequency response lower. To put it simply, a stereo clip doesn't represent all frequencies as accurately as a mono clip. You'll hear the two channels, but they may not sound as crisp as a mono channel. This means that a mono codec may be a better choice for low bit rates.

Stereo music steps into the limelight at higher bandwidths. The higher the streaming speed, the better the stereo frequency response. The stereo music codecs were specifically created to create CD-quality stereo music for broadband and portable music users

About High-Response Codecs

The 20 kbps, 32 kbps, and 44 kbps music codecs come in two flavors. Helix Producer by default uses the "high response" versions, which are the better codecs for most situations. But you can also use the "normal response" versions.

The high response codecs cover a larger frequency spectrum than the normal response versions. Sometimes, the high response version has twice the range as the normal codec. This means it provides crisper sound and is better at capturing high frequencies. With symphonic music, for example, the high response codec gets more of the flute and piccolo. It can produce more distortion than the normal response codec with voices and loud sounds such as drums, though.

If you're encoding music with a diverse range of frequencies, stick with the high response codecs. If you notice distortion, compare your results with a clip that uses the normal response codecs. The best tool for determining which codec to use is your ear. Listen carefully for minute differences in how the clip sounds. It also helps to have other people listen. Our own ears have different frequency responses, too.

The slowest stereo codec is 12 kbps. Stereo codecs don't go lower than that because they would not have enough frequency response for adequate sound. In the following table, the RealAudio 8 codecs are marked with RA8.

About Surround Codecs

At 44 kbps, 64 kbps, 96 kbps, 132 kbps, 146 kbps, 176 kbps, 264 kbps, and 352 kbps bit rates, you have the choice to use RealAudio Surround codecs. These codecs preserve the matrixed multi-channel surround audio in conventional

"surround sound" audio content. Sources of such content are created using Dolby Pro Logic or Circle Surround encoders, and these sources are prevalent on DVDs and television broadcasts.

Surround audio can consist of four sound channels (left, right, left surround, and right surround) or 5.1 channels (additional subwoofer and center) that are mixed into the two conventional left and right stereo channels. Thus, no special cables are needed to encode RealAudio Surround, as long as your audio source is conventional "surround sound" content. While you can use non-digital sources, the best results are achieved from a digital source.

You should only choose the surround versions of the stereo RealAudio codecs if you know that there is surround information in the audio source. The main reason to make sure you have the correct source is that the surround codecs are not as efficient when encoding non-surround audio.

To play back a RealMedia file encoded with RealAudio Surround, you will need to use to an A/V receiver equipped with surround sound decoding that is connected to the surround channels and an optional subwoofer. Your receiver's manual will contain information on how to properly connect surround speakers. Alternatively, some computer speakers (such as Creative Labs Inspire™ 5.1 Digital (5700) speaker system) will play "surround sound" audio.

Mono Music Codecs

As with the voice codecs, the lowest-speed mono music codec normally used with RealAudio is 16 kbps. The lower-speed codecs can be used as "duress" streams in SureStream clips, and to encode soundtracks for low-bandwidth RealVideo clips. When there are two versions of a codec, Helix Producer uses the high response version by default.

RealAudio Mono Music Codecs

RealAudio Codec	Sampling Rate
6 kbps Music	8 kHz
8 kbps Music	8 kHz
11 kbps Music	11.025 kHz
16 kbps Music	22.05 kHz
20 kbps Music	22.05 kHz
20 kbps Music—High Response	44.1 kHz

(Table Page 1 of 2)

RealAudio Mono Music Codecs (continued)

RealAudio Codec	Sampling Rate
32 kbps Music	44.1 kHz
32 kbps Music—High Response	44.1 kHz
44 kbps Music	44.1 kHz
64 kbps Music	44.1 kHz

(Table Page 2 of 2)

JOB FILE REFERENCE

This appendix provides a reference for job files created with Helix Producer. It shows you the basic structure of a job file and gives you details about the different sections within a job file.

Overview: Job Files

Job files are XML-based files that you edit using any text editor. These files can be used with either the main Helix Producer interface or the command line application to customize the settings that you will use for encoding.

Job File Structure

The following is an overview of the structure of the elements within the job file and how they are nested.

job
    enableTwoPass
    clipInfo
        entry >
    input
        avFileInput | captureInput
            prefilters >
    output
        mediaProfile
            postfilters >
            prefilters >
        destinations
            fileDestination >
            pushServer >
            pullServer >
            g2PushServer >
    audiences
        audience
            streams

```
audioStream >
    streamContext >
videoStream >
eventStream >
imageMapStream >
```

A job file's structure can be divided into five main sections: job section, clip information section, input section, output section, and audiences section. Each section and their sub-elements are documented separately below.

Job Section

The top level of the job file is the job section. This section contains all tags within the job file.

enableTwoPass

This setting determines whether 2-Pass encoding is used for the job. Value can be True or False.

Clip Information Section

In this section, the metadata information about the encoded clip is stored. For each entry included, a name/value pair exists. This section is optional.

Structure:

```
clipInfo
    entry
        name
        value
```

entry

Each entry element contains one metadata information as a name/value pair. A number of standard metadata name/value pairs are supported by Helix Producer, but you may add your own.

The following table shows the standard metadata name/value pairs:

Standard Metadata

Name	Value	Description
Title	Any string up to 255 characters in length.	The name of the encoded clip.
Author	Any string up to 255 characters in length.	The author of the encoded clip.
Copyright	Any string up to 255 characters in length.	The owner and date of the copyright.
Keywords	Any string up to 1024 characters in length.	Keywords that enable a user to find the clip.
Description	Any string up to 4095 characters in length.	A brief description about the content of the clip.
Content Rating	Any integer from 0 to 6.	A rating system, where the following values equal the following ratings: 0 = no rating 1 = all ages 2 = older children 3 = younger teens 4 = older teens (15 and up) 5 = adult supervision recommended 6 = adults only

Example:
```
<clipInfo>
    <entry>
        <name>Title</name>
        <value type="string">My Title</value>
    </entry>
    <entry>
        <name>Copyright</name>
        <value type="string">(c) 2002 XYZ Media Corp.</value>
    </entry>
</clipInfo>
```

Input Section

This section contains information about the input for the encoding job. Only one of two elements can be contained in this section, an AV file input section or a capture section.

Structure:

```
input
    avFileInput | captureInput >
```

avFileInput

This sub-section contains properties that define the audio or video file used as the source for the input.

Structure:

```
avFileInput
    pluginName
    filename
    prefilters>
```

The following are the avFileInput properties:

pluginName
Allows you to specify the file reader used to scan the input file. If not specified, all readers are scanned. Use one of:

```
rn-avfile-aviuncompressed
rn-avfile-directshow
rn-avfile-pcm
rn-avfile-qtcompressed
rn-avfile-qtuncompressed
```
or a 3rd party AV file reader

filename
The file used as the source for the input. Use any valid filename, including the path.

prefilters
For more information, see "prefilters" on page 124.

Example:

```
<avFileInput>
    <filename type="string">C:\\media\video1.mpg</filename>
</avFileInput>
```

captureInput

This sub-section contains properties that define the audio or video capture device used as the source for the input.

Structure:

```
captureInput
    prefilters >
    pluginName (rn-capture-av)
    duration (infinite)
    audioDeviceID
    audioDevicePort
    videoDeviceID
    videoDevicePort
    videoFormat
    videoFrameWidth
    videoFrameHeight
```

The following are the captureInput properties:

prefilters
For more information, see "prefilters" on page 124.

pluginName
Allows you to specify the capture driver used. Use rn-capture-av or a 3rd party plug-in.

duration
The length of time that the capture device will be used as the input source. Use the format: [D:][h:][m:]s[.xyz]. Default is infinite.

audioDeviceID
Sets the audio capture device that will be used as the source for the input.

audioDevicePort
Sets the audio port that will be recorded from.

videoDeviceID
Sets the video capture device that will be used as the source for the input.

videoDevicePort

Sets the video port that will be recorded from.

videoFormat

The video transmission format for the source video. Default is NTSC.

videoFrameWidth

The width (in pixels) of the video image that will be captured by the video capture device. Default is 0; the last device setting or default will be used.

videoFrameHeight

The height (in pixels) of the video image that will be captured by the video capture device. Default is 0; the last device setting or default will be used.

Example:

```
<captureInput>
    <duration type="duration">01:20:00.000</duration>
    <pluginName type="string">rn-capture-av</pluginName>
    <videoDeviceID type="string">MSVIDEO: XYZ Capture Driver</videoDeviceID>
</captureInput>
```

prefilters

Prefilters can be used nested within either the avFileInput sub-section or the captureInput sub-section. Using prefilters adjusts the input video or audio data before encoding. They are optional.

Structure:

```
prefilters
    inputCroppingPrefilter
        enabled (true)
        left (0)
        top (0)
        width (input width)
        height (input height)
    deinterlacePrefilter
        enabled (true)
        manual (false)
        deinterlace (true)
        inverseTelecine (true)
    videoNoiseReductionPrefilter
        enabled (true)
        level (low)
    blackLevelPrefilter
        enabled (true)
```

```
audioGainPrefilter
    enabled (true)
    gain (0)
```

inputCroppingPrefilter

Contains settings for cropping the size (in pixels) of the video input. Properties:

enabled

The state of the filter. Value can be True or False.

left

The left side of the cropping region.

top

The top side of the cropping region.

width

The width of the cropping region.

height

The height of the cropping region.

deinterlacePrefilter

Contains settings for De-interlace and Inverse Telecine filters that adjust the input video. Properties:

enabled

The state of the filter. Value can be True or False.

manual

Value can be True or False. If False (default), the filter will apply the De-interlace and Inverse Telecine filter if appropriate. If True, the filters you specify to be used will be used.

deinterlace

Determines if the De-interlace filter is used. Value can be True or False.

inverseTelecine

Determines if the Inverse Telecine filter is used. Value can be True or False.

videoNoiseReductionPrefilter

Contains settings for the reducing video noise from the input video. Properties:

enabled

The state of the filter. Value can be `True` or `False`.

level

Specifies the type of noise reduction used. Value can be `low` or `high`.

blackLevelPrefilter

Contains settings for filtering the black level of the input video. Properties:

enabled

The state of the filter. Value can be `True` or `False`.

audioGainPrefilter

Contains settings for changing the audio level from the input audio. Properties:

enabled

The state of the filter. Value can be `True` or `False`.

gain

The amount of audio gain increased or decreased. Value can be from -12 to 12.

Example:

```
<prefilters>
    <deinterlacePrefilter>
        <deinterlace type="bool">true</deinterlace>
        <enabled type="bool">true</enabled>
        <inverseTelecine type="bool">true</inverseTelecine>
        <manual type="bool">false</manual>
    </deinterlacePrefilter>
    <audioGainPrefilter>
        <enabled type="bool">true</enabled>
        <gain type="double">0.000000</gain>
    </audioGainPrefilter>
</prefilters>
```

Output Section

This section contains information about the output for the encoding job. It includes information about the destinations for the encoded output and how the media is encoded.

Structure:

```
output
    destinations
        fileDestination
            pushServer
            pullServer
            g2PushServer
        mediaProfile
            audienceRef
            audioMode (music)
            videoMode (normal)
            disableAudio (false)
            disableVideo (false)
            outputWidth (Input Width)
            outputHeight (Input Height)
            resizeQuality (high)
            audioResamplingQuality (high)
```

destinations

This sub-section contains information about one or more file or server destinations.

fileDestination

Contains settings for the output file—the encoded RealMedia file. Properties:

pluginName
The name of the file destination plug-in that is used to create the file. Use one of rn-file-realmedia, rn-file-mp3, or a 3rd party plug-in.

name
Name of the output.

filename
The file name and path were the encoded output will be saved.

pushServer

A Push Server destination contains information required for a live server output using the Push broadcast method available in Helix Universal Server 9 and greater. There are three types of Push broadcasts: account-based, single-password, and multicast. The properties you use depends on which type of broadcast are using. For a detailed description of these types of Push

broadcasting and the settings you need, see "Broadcasting Live to a Server" on page 37.

Properties:

name
The name of the server destination.

authType
The type of authentication used for the Push broadcast. Depending on which type you use, you need to use different properties. Use either `account-based` or `single-password`.

streamname
The name of the live stream used to identify the live broadcast. Use any valid file name and path. For example: `live.rm`.

path
Used if you want to use additional path information to define your stream, such as for archiving purposes.

address
The address of the Helix Universal Server being broadcast to. Use any valid IP address.

port
For account-based authType, this is the authentication/monitoring port on Helix Universal Server. For single-password authType, this is the starting port for sending packets, and it must match the receiver configuration defined on the Helix Universal Server.

endPort
The ending port used for sending packets. Used with single-password authType only.

username
The user name of a valid Helix Universal Server account. Used with account-based authType only.

password
For account-based authType, this is the valid password on the Helix Universal Server used in conjunction with the user name. For single-password authType, this is the password used to encrypt and decrypt packets before transmission to the Helix Universal Server.

savePassword
Allows you to save the password if value is `True`. Value can be `True` or `False`.

transport
> The transport protocol used to connect to the Helix Universal Server. Value can be udp/unicast, udp/multicast, or tcp. See "Transport Protocols" on page 43 for more information about protocols.

listenAddress
> The address that Helix Producer uses to listen for resend requests from the broadcasting Helix Universal Server. Use any valid IP address.

multicastAddress
> The multicast address used for the broadcast if transport is set to udp/multicast. Use a valid multicast address.

multicastTTL
> The number of router hops that a multicast stream will cross before being blocked by the router. Limits the spread of a multicast. Value can be any integer from 1-255.

fecPercent
> The percentage of stream allocated to forward error correction packets. If value is 100, a redundant stream is created. Value can be from 1-100.

fecOffset
> The number of seconds used to offset FEC packets when fecPercent is set to 100. Value can be any integer from 1-9999.

metadataResendInterval
> The number of seconds that Helix Producer will wait between sending header packets (packets that contain information about the broadcast) to theHelix Universal Server.

allowResends
> Whether or not to accept/reject requests from the Helix Universal Server to resent packets. Value can be True or False.

enableTCPReconnect
> Whether or not to enable TCP reconnects when the broadcast is dropped by the Helix Universal Server. Value can be True or False.

TCPReconnectInterval
> If enableTCPReconnect is True, determines how long to wait in between reconnect attempts. Value can be any integer from 1-3600.

statisticsUpdateInterval
> The number of seconds in between updates of statistics sent from the Helix Universal Server.

pullServer

A Pull Server destination contains information required for a live server output using the Pull broadcast method available in Helix Universal Server 9 and greater. For a detailed description of Pull broadcasting and the settings you need, see "Setting Up a Pull Broadcast" on page 48.

Properties:

name
> The name of the server destination.

streamname
> The name of the live stream used to identify the live broadcast. Use any valid file name and path. For example: live.rm.

path
> Used if you want to use additional path information to define your stream, such as for archiving purposes.

listenAddress
> The address used by Helix Producer to listen for Pull broadcast connection requests from a Helix Universal Server. Use only a valid IP address.

listenPort
> The port used by Helix Producer to listen for Pull broadcast connection requests from a Helix Universal Server. Value can be any integer from 1-65535.

password
> The password used by Helix Producer to initiate Pull broadcast requests. Must match the Broadcast Distribution Pull Receiver Password set in Helix Universal Server.

savePassword
> Allows you to save the password if value is True. Value can be True or False.

serverTimeout
> The number of seconds Helix Producer will wait for a ping from the Helix Universal Server before assuming that no clients are connected to the stream, and then close the broadcast connection. Value can be any integer from 0-86400.

g2PushServer

A G2 Push Server destination contains information required for a live server output using the Legacy broadcast method available with older versions of Helix Universal Server. For a detailed description of Pull broadcasting and the settings you need, see "Setting up a Legacy Push Broadcast" on page 50.

Properties:

name
The name of the server destination.

streamname
The name of the live stream used to identify the live broadcast. Use any valid file name and path. For example: live.rm.

path
Used if you want to use additional path information to define your stream, such as for archiving purposes.

address
The address of the Helix Universal Server being broadcast to. Use any valid IP address.

port
The port on Helix Universal Server being broadcast to. It should match the encoder port setting on Helix Universal Server.

username
The user name of a valid Helix Universal Server account.

password
This is the valid password on the Helix Universal Server used in conjunction with the user name.

savePassword
Allows you to save the password if value is True. Value can be True or False.

transport
The transport protocol used to connect to the Helix Universal Server. Value can be udp/unicast or tcp. See "Transport Protocols" on page 43 for more information about protocols.

mediaProfile

This sub-section defines settings for how the output is encoded, plus it defines the audiences used for encoding.

Structure:

```
mediaProfile
      audioMode (music)
      videoMode (normal)
      disableAudio (false)
      disableVideo (false)
      outputWidth (InputWidth)
      outputHeight (InputHeight)
      resizeQuality (high)
      audioResamplingQuality (high)
      audienceRefs
          audienceRef
```

Properties:

audioMode
> Describes the type of audio included in the encoded clip. The choice here affects the type of audio codec used to encode the output. Value can be either voice or music.

videoMode
> Describes the type of video content included in the encoded clip. This setting controls the target quality of the encoded video, based on your choice. Value can be sharp, normal, smooth, or slideshow.

disableAudio
> If False then audio is disabled for the output. Value can be True or False

disableVideo
> If False then video is disabled for the output. Value can be True or False

outputWidth
> The width of the output video image in pixels. A value between 32 and 2048 results in the output being scaled to fit. If value is 0 and outputHeight is 0, then no resizing occurs. If value is 0 and height is between 32 and 2048 then width is computed such that the aspect ratio of the input video is maintained.

outputHeight
> The height of the output video image in pixels. A value between 32 and 2048 results in the output being scaled to fit. If value is 0 and outputWidth is 0, then no resizing occurs. If value is 0 and width is between 32 and 2048 then height is computed such that the aspect ratio of the input video is maintained.

resizeQuality

Affects the resulting video quality of an output file when resize is applied. Choosing High results in a better quality video but uses considerably more CPU. Value can be High or Fast.

audioResamplingQuality

Affects the resulting quality of audio resampling operations. Resampling must be done if the sample rate required by the target audio codec is different from the input file or device sample rate. Value can be High or Fast.

audienceRefs

This property holds a reference to the name of an audience defined in the audiences section of the job file. One or more of this tag may appear. Each occurrence represents one audience that is to be added to the mediaProfile.

Example:

```
<mediaProfile>
    <audioMode type="string">music</audioMode>
    <audioResamplingQuality type="string">high</audioResamplingQuality>
    <disableAudio type="bool">false</disableAudio>
    <disableVideo type="bool">false</disableVideo>
    <outputHeight type="uint">0</outputHeight>
    <outputWidth type="uint">0</outputWidth>
    <resizeQuality type="string">high</resizeQuality>
    <videoMode type="string">normal</videoMode>
    <audienceRefs>
        <audienceRef>16k Substream for 28k Dial-up</audienceRef>
        <audienceRef>28k Dial-up</audienceRef>
        <audienceRef>56k Dial-up</audienceRef>
        <audienceRef>256k DSL or Cable</audienceRef>
    </audienceRefs>
</mediaProfile>
```

Audiences Section

This section defines one or more audiences that are used to encode the output media. For more detailed information on audiences, see Chapter 7: "Choosing Audiences for an Encoding Job" on page 55.

Structure:

audiences

```
audience
    name (only within job file)
    avgBitrate
    maxBitrate
    streams
        audioStream >
        videoStream >
        eventStream >
        imageMapStream >
```

name

The name assigned to an audience. This name must match one of the audiences listed in the audienceRef property of the mediaProfile section.

avgBitrate

Bit rate in bps (bits per second) of the entire audience. Ignored if encodingType is quality.

maxBitrate

Limits total bit rate for the audience from going above the specified value in bps (bits per second). Set to 50000000 or more if no limit is desired.

streams

This sub-section is a container for the video stream and the audio stream(s) included with the audience.

Structure:

```
streams
    audioStream
        codecName
        codecFlavor
        streamContext
            presentationType (audio-video)
            audioMode (music)
    videoStream
        codecName (rv9)
        encodingType (cbr)
        quality
```

```
            maxStartupLatency (4 s)
            maxFrameRate (30 fps)
            maxKeyFrameInterval (10 s)
            enableLossProtection (false)
        eventStream
            avgBitrate (200 bps)
        imageMapStream
            avgBitrate (400 bps)
```

audioStream

Defines settings for an audio stream. Properties:

codecName

The name of the audio codec to be used. See "Music RealAudio Codecs" on page 104 and "Voice RealAudio Codecs" on page 106 for a list of values.

codecFlavor

The flavor of the audio codec to be used. See "Music RealAudio Codecs" on page 104 and "Voice RealAudio Codecs" on page 106 for a list of values.

streamContext

Defines a unique set of properties to allow the application to select only one audio stream among several choices. The following are sub-properties:

presentationType

Describes the type of presentation (if it is audio only or audio and video). Value can be audio-only or audio-video.

audioMode

Describes the type of audio content. Value can be voice or music.

Example:

```
<audioStream>
    <pluginName type="string">rn-audiocodec-realaudio</pluginName>
    <codecName type="string">cook</codecName>
    <codecFlavor type="uint">4</codecFlavor>
    <streamContext>
        <presentationType type="string">audio-video</presentationType>
        <audioMode type="string">music</audioMode>
    </streamContext>
</audioStream>
```

videoStream

Defines settings for a video stream. Properties:

codecName

The RealVideo codec name used to encode the video. Below is a table that lists codec names and the corresponding RealVideo codec.

Table 4: RealVideo Codecs

Name	RealAudio Codec Used
rv9	RealVideo 9
rv8	RealVideo 8
rvg2svt	RealVideo G2 with SVT

encodingType

The type of encoding used can be Constant Bit Rate (**cbr**), Variable Bit Rate using quality of video as a goal (**vbrQuality**), or Variable Bit Rate using an average bit rate as a goal (**vbrBitrate**). If you use Variable Bit Rate, you can only use one audience and the output file name must use a **.rmvb** extension.

quality

The target quality percentage that the video codec will attempt to encode the video at. Used if you specify the **encodingType** to be **vbrQuality**.

maxStartupLatency

The maximum time that the clip will buffer before playing begins. Value can be an integer from 4 to 25.

maxFrameRate

The ideal, maximum frame rate (in frames per second) that the video codec will attempt to encode the output video at. Value can be an integer from 0 to 60.

maxKeyFrameInterval

The number of seconds in between key frames of encoded video. Value can be an integer from 0 to 60.

enableLossProtection

Whether or not to add error correction information to the encoded file. Value can be True or False.

Example:

```
<videoStream>
   <pluginName type="string">rn-videocodec-realvideo</pluginName>
   <codecName type="string">rv8</codecName>
   <encodingType type="string">cbr</encodingType>
```

```
<maxStartupLatency type="double">4.0</maxStartupLatency>
<maxFrameRate type="double">15.0</maxFrameRate>
<maxKeyFrameInterval type="double">10</maxKeyFrameInterval>
<enableLossProtection type="bool">false</enableLossProtection>
</videoStream>
```

eventStream

Defines settings for an event stream. Properties:

avgBitrate
Bit rate for event stream in bits per second. Value can be from 1-1000.

imageMapStream

Defines settings for an image map stream. Properties:

avgBitrate
Bit rate for image map stream in bits per second. Value can be from 1-1000.

INDEX